THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR T. Craig Montgomerie

TITLE OF THESIS INSTRUCTIONAL DEPARTMENT COMPUTER SUPPORT SYSTEMS: A STRATEGY AND AN IMPLEMENTATION

DEGREE FOR WHICH THESIS WAS PRESENTED DOCTOR OF PHILOSOPHY

YEAR THIS DEGREE GRANTED 1981

THE UNIVERSITY OF ALBERTA

INSTRUCTIONAL DEPARTMENT COMPUTER SUPPORT SYSTEMS:
A STRATEGY AND AN IMPLEMENTATION

by

© T. Craig Montgomerie

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

IN

EDUCATIONAL ADMINISTRATION

EDMONTON, ALBERTA

SPRING, 1981

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled

INSTRUCTIONAL DEPARTMENT COMPUTER SUPPORT SYSTEMS:

A STRATEGY AND AN IMPLEMENTATION

submitted by T. Craig Montgomerie in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Educational Administration.

DEDICATION

To my daughter Lindsay Jaye Montgomerie.

ABSTRACT

This study consists of the development of a strategy for an educational administrator to follow when directing the implementation of an instructional department computer support system. The strategy is tested by actual application on a real problem: the design and implementation of the Department of Educational Administration Computer Support System (DEACSS).

The major objectives of the study were:

1. the development of a strategy for the design and implementation of an instructional department computer support system,
2. testing of this strategy by using it to develop a fairly large, in both scale and scope, computer support system - the Department of Educational Administration Computer Support System (DEACSS),
3. provision of a set of cost figures, subdivided into different task categories, for the implementation of DEACSS,
4. evaluation of how well DEACSS met the needs of the department, and
5. evaluation of the strategy used to develop DEACSS and the drawing of implications concerning the general strategy which would assist other middle level administrators wishing to implement a similar system.

All five of these objectives were met. DEACSS was evaluated as being successful in meeting the information and support needs of the Department of Educational Administration. The strategy was evaluated as successful within the restrictions of this single application.

ACKNOWLEDGEMENTS

This thesis has been the culmination of a great deal of time, effort and suggestions from many people over a number of years.

Principle acknowledgment must go to Dr. J.E. Seger who has been everything one could ask for as a supervisor, teacher, collaborator and friend. Dr. S.M. Hunka, for whom I have worked during the development of this project, has provided me with the freedom to work on the project as needed, and by example, has shown me the real meaning of academic freedom.

The other members of my committee, Dr. D. Richards, Dr. C. Bumbarger, and Dr. W. Muir have given me many positive suggestions and a great deal of support.

Ron Senda, Marc LaFrance and Paul Kam of the University of Alberta and Dick Guertin of Stanford University have provided me with a tremendous amount of free assistance when SPIRES refused to work as I thought it should. In most cases the fault was my own, but in those cases where SPIRES was at fault, they were very quick to provide either changes to SPIRES or to suggest alternative ways to accomplish what I needed.

Judy McKinney was Department Secretary to the Department of Educational Administration during the specification and development of DEACSS. She provided invaluable assistance in advising what should be included in

DEACSS, and as the person who entered, checked and corrected most of the data in DEACSS.

Deborah Dancik edited this thesis. As well as correcting the English and helping to arrange concepts into some rational order, she constantly questioned my basic assumptions which helped to focus my thoughts and, hopefully, my prose.

Finally, I must acknowledge the support of my wife Heather and my daughter Lindsay Jaye. Without their understanding and support I could never have completed this task.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

Electronic digital computers have now been in existence for about thirty five years. In this time, the use of the electronic digital computer (from this point on referred to as a "computer") has developed from its first intended application[1] to the promise of the micro computer as

> "a universal personal accessory that will be more important in our daily lives than the clock, the telephone, the typewriter, television, the calculator, the recorder, the copier, the checkbook, the camera, mail, books, or files, because it will replace all of these things." (Roland, 1979, p. 84).

Currently in the field of education, computers are used extensively in the administration of large institutions and large school systems. Applications range from such mundane tasks as the generation of monthly cheques, to very complex simulations of entire financial systems.

Traditionally, the cost of computer hardware is one area where the economy of scale is exceedingly well demonstrated[2]. As the capabilities of computers have grown over the years, many computer users have come to accept the maxim "bigger is better," not only for computer hardware but, perhaps mistakenly, for software and applications as well.

----

[1] ENIAC, generally accepted as the first modern electronic digital computer, was designed during World War II to calculate shell trajectories. ENIAC became operational in 1946.

[2] An examination of Solomon's 1966 article "Economies of Scale and the IBM/360" or of current price lists from mini and macro computer manufacturers such as the 1979 *PDP11 Systems and Options Summary* from the Digital Equipment Corporation demonstrates this contention quite well.

These capital equipment cost benefits have led to more and more centralization of computing systems within organizations. There has been an associated centralization of software and applications, especially in financial and administrative services. Such centralization has meant that a middle level administrator must accept the computer services offered by the central administration. The interest of the central administration when developing systems is, naturally, to meet its own needs first. As will be shown later, these needs do not always coincide with the needs of the middle level administrator. Currently, the middle level administrator must keep data or files not provided by the central administration's computer service in traditional file folders in file cabinets. This quite often leads to a duplication of data and information as once a file folder is set up, the tendency is to make it as complete as possible so that data on one topic do not need to be collected and collated from many sources.

Even though many middle level administrators may feel frustrated with the limitations of a global system provided by the central administration, in the past they have not had the financial, manpower, organizational or knowledge resources to develop a computer support system (CSS) for their own applications.

Two recent developments in the computer field have led some middle level administrators to investigate the possible introduction of computer technology into their own day to

day tasks:

1. the development of low cost micro computer systems, and
2. the thrust of systems designers working on large (macro) computers towards the development of more easily used, human engineered systems[3].

Most middle level administrators however are not skilled computer analysts or programmers (Plice, 1980, p. 285). Many middle level administrators have never taken a formal computer course[4]. They are not certain just what they can expect a computer to do for them, what to ask for, or even where to start.

1.1 Purpose of the Study

The purpose of this study was to develop a strategy for the design and implementation of an instructional department computer support system (IDCSS). This strategy would be tested by using it to design and implement an IDCSS to provide assistance for many of the diverse needs of an instructional department at the University of Alberta. An instructional department can be classified as middle level

----

[3] Human engineering refers to the engineering of equipment and systems in such a way that the technology adapts to the needs of humans, rather than humans adapting to the needs of technology. Gruhn and Hohl discuss many of the features of the human engineered computer systems currently used in the IBM Thomas J. Watson Research Center.

[4] A recent survey of students enrolled in the Department of Educational Administration (Montgomerie 1981) revealed that only 41% of these students had a formal course concerning computers. One requirement for entrance to this department is that the student have some administrative experience.

administration, with many of the department information needs far removed from those of the central administration. The IDCSS to be developed would be both economical to institute and maintain as well as sufficient to meet the immediate and future needs of the department. Further, the strategy used to develop the IDCSS would be generalizable to other instructional departments both within this university and other institutions.

This study is comprised of five parts:

1. the development of a strategy for the design and implementation of an instructional department computer support system (IDCSS),
2. testing of this strategy by using it to develop a fairly large, in both scale and scope, computer support system - the Department of Educational Administration Computer Support System (DEACSS),
3. provision of a set of cost figures, subdivided into different task categories, for the implementation of DEACSS,
4. evaluation of how well DEACSS meets needs of the department, and
5. evaluation of the strategy used to develop DEACSS and the drawing of implications concerning the general strategy which would assist other middle level administrators wishing to implement a similar system.

1.2 Definitions of Terms Used

In this thesis there are technical terms which have specific meanings in the area of computer support systems (CSS). The majority of these technical terms are from the areas of management information system (MIS), data base management system (DBMS) and the Stanford Public Information Retrieval System (SPIRES). These terms will be defined as they are introduced in the text. A summary is attached as Appendix B - A Glossary of Terms.

Since they are used in the following discussion, a few terms must be defined at the outset:

DATA: A set of characters, words or signals to which a significance can be assigned. (Hussain, 1973, p. 81)

INFORMATION: Selected data that have been processed to make them meaningful. (Hussain, 1973, p. 81)

DATA BASE: A collection of data organized to facilitate maintenance, query, and/or reporting.

DATA BASE MANAGEMENT SYSTEM: A method of managing and manipulating the data in a data base.

MTS: An acronym for **M**ichigan **T**erminal **S**ystem, the Operating System which controls the execution of jobs on the computer at the University of Alberta.

SPIRES: The **S**tanford **P**ublic **I**nformation **RE**trieval **S**ystem. A generalized data base management system designed to be accessed primarily for on-line applications.

1.3 An Overview of the Study

This study consists of the development of a strategy for an educational administrator to follow when directing the implementation of an instructional department computer support system. The strategy is tested by actual application on a real problem: the design and implementation of the Department of Educational Administration Computer Support System (DEACSS).

Chapter two establishes the perspective for the study. It contains an introduction to data management on computers, discussions of data base management systems, Management Information Systems, and a discussion of Computerized Management Information Systems in the field of Education.

Chapter three defines and specifies the attributes of, as well as detailing a strategy for the design and implementation of, an instructional department computer support system. Chapter four discusses the use of this strategy to design and implement the DEACSS System. Chapter five includes a cost accounting for the implementation and maintenance of DEACSS in terms of man hours and computer costs. Chapter six consists of an evaluation of DEACSS and the general strategy used to implement DEACSS. The final chapter summarizes the study, discusses applications of the general strategy used, discusses the implications of the design and implemetation of an instructional department computer support system and suggests some areas for further research.

2. A PERSPECTIVE

Chapter one briefly mentioned some of the background and beliefs which prompted this investigation. This chapter will place the investigation in perspective with respect to the use of computers in business administration and in educational administration. The following major topics in the evolution of computer technology are discussed:

1. methods of data management
2. management information systems
3. applications of management information systems in educational administration
4. SPIRES - an example of a data base management system
5. decision support systems.

This discussion will be directed towards what data base management systems and management information systems are, what they do, and where they have been applied in Education. Areas in which management information systems have succeeded and/or failed will also be examined.

2.1 Data Storage and Manipulation

Data are defined by Hussain (1973, p. 81) as "a set of characters, words or signals to which a significance can be assigned." Data have no inherent meaning by themselves, but are simply a representation of something.

Information is defined by Hussain (1973, p.81) as "selected data that have been processed to make them

meaningful." The intent of many computer programs is to convert data into information. Computer programs can only read, store, retrieve and manipulate data. This process is call data management. It takes the ingenuity of the computer programmer to manage the data in such a way that when the data are retrieved they have been transformed into information.

2.1.1 Traditional Data Management

When digital computers first were designed and used, they could process only a single task or "program" at a time. These computers worked in a very simple manner:

1. The computer program was loaded into the computer.
2. The address at which the program should begin execution was set up on switches on the "operator's console" - the area from which the computer operator controlled the computer.
3. The operator depressed a switch to have the computer begin executing instructions at the address shown on the operator's console.
4. The computer program would read the data it needed from some input device (usually a card reader or paper tape reader).
5. The computer program would make some calculations on the input data to produce information.
6. The computer program would print the information on some output device (usually a printer or a teletype).

7. The computer program would stop.
8. The computer operator would load another program into the computer and the process would begin again.

In order to provide optimum speed, each program would be designed so that the data and only the data required by each program would be presented to that program, arranged in an optimum manner for the processing of that particular program.

This approach led to a tremendous redundancy of data because the same data were stored separately for use with each different program. It could also lead to errors, for instance in cases where the same data in slightly modified forms were maintained for use with different programs, users of the system might update or correct one set of data while forgetting to update the others.

This traditional data management is termed by Cohen (1979, p. 1-2) as "program oriented". Most of the first uses of computers were numerical applications, and many of those were of the type where the data were used only by a single program. In these cases, the traditional data management approach was and still is appropriate.

2.1.2 Data Base Data Management

As computers evolved, the manner in which they worked also evolved. By the mid 1960's, computers were able to process many tasks at the same time by the method known as time sharing whereby multiple tasks would be in some state

of execution in the system at the same time, with some master task allocating time and resources to each task in turn.

In the 1960's, applications of computer technology had moved from almost exclusively numerical areas to areas using data which were not always numerical. Also, as businesses began to use computers, they found the need to access the same data from not one or two programs, but from many programs. The need for more than one program to access the same data forced a reappraisal of the way data were handled. No longer could specific data be stored in a form unique to each program in which they were required. Instead, data needed to be stored in some general form that was easily accessed and manipulated from all programs. This general storage of all data is called a data base.

This concentration on the data itself, rather than on the program which was processing the data, led to what Cohen (1979, p. 1-2) calls the "data base systems approach (which) is *data* oriented rather than *program* oriented."

The data base approach is especially applicable where a great deal of common data must be addressed by a number of different programs. Since many different programs address the same data, care must be taken that the data base be structured for generality, flexibility and extensibility (Korenthal, 1978 p. 4(1):2). These qualities are the exact opposite of the qualities of traditional data management.

The major job of managing the data in a data base is accomplished by what is called a data base management system (DBMS). In an introduction to data base concepts, Korenthal (1978, p. 4(1) 2-3) identifies the following tasks which a DBMS must perform:

1. It must *maintain* the data in the database to allow the adding, modifying and deleting of data.
2. It must provide a *query* function to allow searching of the data base for records which satisfy certain criteria.
3. It must provide a *reporting* facility to allow production of reports on the data found. Reports may be as simple as stating the number of records which satisfy a single criterion or as complex as a statistical analysis of the data associated with all the records which satisfy a combination of many criteria.

2.1.3 Traditional Data Management vs. Data Base Data Management

Traditional data management and data base data management each have valid applications. It is unreasonable to state that data base data management is always superior to traditional data management. It is equally unreasonable to attempt to show that traditional data management is always superior to data base data management. There are however individual applications in which one type of data management should be used in preference to the other.

Traditional data management is more often used when working with data which are applicable to a single task. Data base data management should be used when dealing with "a growing community of data designed to serve a growing community of users" (Cohen, 1979, p. 1-3).

Data base data management traditionally has major costs which make its use on small or single use tasks inadvisable. Cohen suggests that

> "the initial investment in the data base system design and implementation, and in the initial data base itself have their pay-offs and justifications. But these will not be evident in the first application, nor in the second or third, but rather at some point of overall system growth and development. If the data base system turns out to be successful, there will be some point in time at which it begins to return multiples of the effort expended." (Cohen 1979, p. 1-3)

2.2 Management Information Systems

In 1969 J.D. Aron defined a management information system (MIS) as "an information system which provides the manager with the information he needs to make decisions" (Aron, 1969, p. 213). Given Aron's definition of a MIS, it can be seen that a MIS can exist in a purely manual form. A clerk tabulating the number of students within a certain program who are taking different types of courses may be providing management information services to an administrator who is trying to forecast the number of sections of a course to offer in future years.

While it is possible to have manual Management Information Systems, it has become accepted within the computer industry that MIS means computerized MIS. Computerized MIS are usually built upon a DBMS (Cohen, 1979, p. 1-2). While the two names both contain the words "management" and "system", they have two very different meanings. The task of a DBMS is to manage data. The task of the MIS is to provide information to management. This difference cannot be emphasized enough, as quite often professionals in the area of DBMS confuse one with another. Some think that by managing the data in an expert fashion, they have somehow created a MIS (Keen and Scott Morton 1978, p. 54). Section 2.3 discusses this problem.

A computerized MIS relies upon the DBMS to provide the maintenance, query and reporting facilities. The MIS provides the "intelligence" to form the data received into meaningful information.

The success or failure of the MIS depends to a great extent on the original design of the data base. Schwartz (1970, pp. 28-31) developed a twelve step approach to planning the design and implementation of a MIS. In the procedure, Schwartz proposed that a great amount of time should be spent finding out what data to include in the data base, and identifying the "broad missions and specific objectives" (Schwartz 1970, p. 28) for the data base. This identification should come from management, line and staff people whose "ideas at many points are likely to be more

valuable than those generated by systems technical experts." (Schwartz, 1970, p. 30). This was to be done before any computer programming was undertaken. As Weizenbaum (1976, pp. 111-131) has emphasized, often system designers are too interested in seeing concrete results (i.e. computer code), rather than spending the appropriate amount of time making sure that the data base will meet the needs of the users once it it implemented.

Schwartz also points out that the generation of a MIS design must be an evolutionary procedure. The first design will not be perfect, so that the "planning process must be concurrent with the implementation process" (Schwartz, 1970, p. 30). This supports the contention that the design of any system must not only be amenable to but in fact must plan for change.

2.3 Criticisms of Current Management Information Systems

Nine years after Aron defined MIS, Keen and Scott Morton made the following comment on MIS: "there is no generally accepted definition recognized by those working in the field." (Keen & Scott Morton, 1978, p. 33). This is attributed to the fact that "there is not yet a mature academic MIS field of study which would stand as a 'discipline' in its own right." (Keen & Scott Morton, 1978, p. 34).

This lack of a formal discipline of MIS is further

complicated by the fact that there are two major kinds of professionals involved in MIS systems:

1. the managers, who may have only a cursory knowledge of the technical concepts involved in MIS, but who are "ultimately responsible for the success of information systems".
2. the data processing professionals who "tend to view it in terms of programs or functions." (Keen & Scott Morton, 1978, p. 34).

As has been stated earlier, many data processing professionals perceive a MIS purely as a data processing problem - not as a problem of providing specific information to improve management decisions. Some data processing professionals are beginning to realise the fallacy in this assumption. Tsichritzis and Lochovsky (1979, p.117) criticize many current practices in the design of Data Base Systems.

> "A Data Base Management System can be an effective data management tool, provide invaluable help in coping with data organization and access problems, and improve the quality of information available for management decision making. Or it can be an inflexible and costly addition to the dp[5] budget, providing management with more headaches than help. The difference is often determined by how the data bases are generated.
>
> Data base generation, the process of determining the data organization and processing requirements of an enterprise, generating a suitable description of these in terms of a schema, and converting existing files and programs according to the schema, is the critical first step in adopting the DBMS approach. Data base generation is often treated as merely a file conversion problem. Data

----

[5] Authors note: data processing.

> are converted with little or no analysis of the appropriateness of their organization. The result is that the DBMS is fitted to the *existing* processing environment and is used as a glorified access method.
>
> Data base generation should not be regarded as a conversion problem, but as an opportunity to plan the organization, use, and management of data. The emphasis should be on analyzing the data requirements of a business or other enterprise, and on the accurate reflection of these requirements in the schemas."

Following this criticism of some current practices, the authors provide an alternative approach to implementation.

> "The first and most important step in data base generation is to determine the data organization requirements of the different components of the enterprise...Unfortunately, this step has been largely ignored or given only cursory attention...Data organization requirements are best identified by conducting a series of interviews within the various *user* departments." (emphasis added to original)

Holland (1980) reinforces the argument of the necessity to obtain user views. He argues that data base developers ask many technical questions, but "spend no time asking equally important questions regarding the intended use of the data."

Holland says:

> "We are interested in obtaining the data requirements of one or more applications, translating these requirements into one or more sets of subschema, then merging the subschema into a combined set called a schema. The schema then is a logical representation of a combined set of user requirements that we call user views."

but that

> "Companies ... have generally found difficulty in acquiring user views. The primary factors of designer concerns are education of users, definition of a user view, gathering user views, complexity of

> views, and integrating user views." (Holland, 1980, p. 141).

In 1978, the theme of the twenty-third annual College and University Machine Records Conference (CUMREC) was "User Information Systems". The conference logo was "Management Information Systems" with the word "Management" crossed out and the word "User" substituted. Richard and Richard (1978, p. 117) comment on this logo:

> "It managed to convey, quite clearly, the idea that User Information Systems are to be the Phoenix like reincarnation of abortive attempts to conceive, design, and implement Management Information Systems."

In their claim that Management Information Systems have failed to serve the needs of the users, Richard and Richard itemize many of the failings of early MIS's. They identify one major failing as the inability to implement the "total systems approach", a widely held concept in the early 1970's that MIS would ultimately provide full range support for *all* levels within organizations. The total systems approach was a reaction to early, single application, computerized systems which had been developed by each user group to meet their own specific needs. The Total Systems Approach attempted to vertically integrate the information needs from all levels in an organization into a single, monolithic system.

Richard and Richard (p. 117) indicated that it is impractical or impossible for a single system to meet the needs of all users in an organization equally well, since

the requirements of users for the reporting and combining of information vary so greatly between different levels of organizational hierarchy. The requirements of the "upper levels" are for information general enough to allow extrapolation and investigation of "what if" type situations. The "lower levels" on the other hand need to be able to access specific information that guide specific operational decisions.

Richard and Richard proposed that each level of user must be able to maintain a data base in the most appropriate format. They proposed and developed the "Generalized File Synthesizer" which allows a given user to access the elements in any existing data bases in order to recover the information needed by that particular user, and to merge this with that user's own data base.

2.4 Applications of Management Information Systems in Educational Administration

A recent article was entitled: "Information Systems and Educational Administration: Totally Inseparable and Generally Archaic" (Simmons 1979, p.16). This article makes a rather scathing attack on the current state of the lack of use of information systems in educational administration. Simmons compares the acceptance of computer technology with earlier technologies.

> "Education has traditionally been a late adopter of new technologies. Such technologies as film,

> television, and programmed instruction have usually originated in military or industrial research and development settings, been incorporated by the business and industrial sectors of our economy, and then, slowly, moved into the mainstream of the educational complex. Generally, such additions to the public instruction scene have been accepted and utilized with positive effects, albeit with frequent misuse of the potentials of such innovations. Computer technology has been incorporated into public education via the same route. However, whereas earlier technologies have generally failed to exert other than a peripheral influence upon educational administration and processes, it can be demonstrated that computer technology is currently a central and critically important factor at all levels of operations in urban school districts throughout the United States." (Simmons, 1979, p. 16)

Simmons also points out in a footnote that discussions with education offices at both the federal and state level would indicate that the improved use of information systems is not an important consideration to these agencies.

There are reasons why educational administration has been slow to adopt the use of information systems technology, but there are indications that educational institutions are now moving towards the use of MIS.

The use of DBMS and MIS has not been as rapid or as widespread in education as would be preferred by some people. Some reasons advanced for this are that, until recently, DBMS's were only available on large mainframe computers (Holsapple, 1980, P. 1), and traditional DBMS's require a great deal of expensive manpower to support the system (Collins, Feeney and Gosden, 1979, p. 59). These kinds of resources were available only to the larger urban school districts or consortia of smaller districts. However,

some of these school districts have made effective use of DBMS and MIS. Hansen, Klassen and Lindsay (1978, pp. 1-10) reviews one of the major consortia - Total Information for Educational Systems (TIES) - which has developed many programs for use of computers in school and school district administration.

2.4.1 MIS in Administration of Post Secondary Education

Colleges and Universities are moving towards the use of Data Base Management Systems and Management Information Systems. An analysis of the 48 papers presented at the 25th Annual College and University Machine Records Conference (CUMREC) showed that 17 were concerned with the design, use, or theory of either DBMS or MIS in educational administration. The distribution of these articles was:

Student Record Systems	5
Theoretical use of M.I.S.	4
Staff/Personnel Record Systems	3
Integrated Systems (Student/Financial)	2
Student Assistance Systems	1
Accounting Systems	1
Purchasing Systems	1

Some authors, Kropf (1980, pp. 61 - 73)in particular, were careful to point out that their MIS was <u>not</u> operating under a DBMS but was a standard data processing system. In the Kropf article, it was apparent that no DBMS was available for the computer on which their MIS was

implemented as their computer system was too small.

2.4.2 A Problem With Current MIS Implementations in Education

One major problem with the design of current MIS's in educational administration might be best defined by two questions:

Who is specifying the system?

Who is using the system?

Earlier, Schwartz (1970, p. 30) was noted as stating that management, line, and staff all should have input into what should be kept in the data base. Broadening this idea, the people who will be using the system should have input into what the system should contain and how it should be input to the system and in what manner they wished to receive output from the system.

In the previously mentioned CUMREC 78 - "User Information Systems" conference, approximately fifteen papers addressed student records and student information systems. The majority of these papers indicated that the student record file was designed and implemented by the computer centre staff and was based upon existing systems (manual or computerized). A few papers indicated that some people in the central administration (notably those in the Registrar's office and those with the word "President" in their title) provided input as to the kind of information to be maintained. None of the papers indicated that anyone

outside of the central administration had been contacted for input, although two articles mentioned Deans as possible users of the information in the DBMS. While users in faculties and teaching departments may not be seen as the most important users of student record systems, Holland (1980, P. 141) and Schwartz (1970, p. 30) would both suggest that they should be consulted on their particular needs, as well as on the form and format of the data.

The most important element in the successful implementation of the next generation of user information systems appears to be the involvement of all levels of users, in an attempt to design a system which will provide the required service to all. The data must be maintained in the most specific form, so that the administrators dealing at the operational level can access the data they need. The DBMS must also be powerful enough to find and combine this specific data in a form which will provide information for use in policy-type decision making.

2.5 Stanford Public Information Retrieval System

The Stanford Public Information REtrieval System (SPIRES) is the major DBMS supported on the University of Alberta's general academic computer system. Since the implementation of the system proposed and designed in this thesis relies on a DBMS, SPIRES was the system which was first recommended, and which turned out to have many

features which made this study work more easily than could have been the case if some other DBMS had been used.

Jackson and Davis (1976, p. 4) give the following as the reason for and the philosophy of SPIRES:

> "A system that can handle diverse information applications must be generalized. That is, the user must be able to declare what type of structure the collection should have, and then to maintain, update, search, and display information from this collection. A generalized system that can handle many types of information structures is also a "cost effective" system; the cost of developing SPIRES is spread over its many users, whereas the cost of developing a dedicated system for one particular application falls entirely on that particular application."

This philosophy led to the development of a very generalized DBMS which was designed so that non-computer professionals could specify, implement, and use their own information system.

2.5.1 The Organization of Data in SPIRES

The organization of data in SPIRES is not unlike the organization of data in many DBMS's. Data elements are organized in a hierarchical arrangement which shows each data element's relationship to each other data element. The following describes the organization of data from the single non-divisible data element up to the total data base - the SPIRES file.

2.5.1.1 The Data Element

The data element is a single value or character string

which cannot be further subdivided. Data elements are the lowest order construction in SPIRES. Eventually all data must reside in a data element.

Examples of data elements are sex, marital status, telephone number or street address.

2.5.1.2 The Data Structure

Data structures indicate hierarchical relationships between data elements or other data structures. Data structures never take an actual value themselves. As an example, an address file might contain addresses and phone numbers for many people. Some people might have both a work address and a home address. In order to keep the data in the file properly organized, the data elements for type of address, street address, city, province and telephone number would be included in a structure. An example of the organization for such a structure is shown in Figure 2.1. This arrangement allows the storage of data elements in relation to each other. Figure 2.2 shows how two occurrences of the address structure keep appropriate information structured together.

Without the ability of the structure to indicate which data elements are related to each other, we could have two street addresses, two cities, two telephone numbers, etc. and not know which street address referred to which city, or which telephone number was the home number and which the work number. Such formatting of data is referred to as a

Figure 2.1 Structure and Elements for Individual Address

- Student Identification Number (Data Element)
- Address* (Structure)
  - Street (Data Element)
  - City (Data Element)
  - Province (Data Element)
  - Country (Data Element)
  - Postal Code (Data Element)
  - Telephone (Data Element)
  - Address Type (Data Element)
- Social Insurance Number (Data Element)
- Birth Date (Data Element)

*May occur more than once.

Figure 2.2 A Structured Arrangement of Two Addresses

Student Identification Number = 631791

Address*
- Street = 11647 - 77 Ave.
- City = Edmonton
- Province = Alberta
- Country= Canada
- Postal Code = T6G 0M4
- Telephone = 403-436-2628
- Address Type = Home

Address*
- Street = University of Alberta
- City = Edmonton
- Province = Alberta
- Country = Canada
- Postal Code = T5G 2G5
- Telephone= 403-432-3762
- Address Type= Business

Social Insurance Number= 606 268 894

Birth Date = Sept. 14, 1945

flat record. Figure 2.3 shows a flat record arrangement of the address data for the two addresses; note that it is impossible to tell, for example, which telephone number belongs to which street address.

2.5.1.3 The Data Record

All the data for one case (person, subject, item) together forms the data record for that case. For example, a student record file might contain data of a personal nature, background information, current registration information and course marks. Together, all the data for one student would form that student's data record.

2.5.1.4 The Subfile

All the data records of a certain type (e.g. student data records) form a subfile. The subfile is given a name usually representative of the type of data it contains.

2.5.1.5 The File

All the data being stored by one application in SPIRES together forms the SPIRES file. A SPIRES file is composed of separate subfiles. Data held in one subfile are totally independent of data held in another subfile unless the two files are explicitly linked together.

For example, a SPIRES file might consist of two completely separate subfiles: a staff file and a student file. These two files would exist in complete isolation from

Figure 2.3 A Flat File Arrangement of Addresses

- Student Identification Number (Data Element)
- Street = 11647 - 77 Ave.
- Street = University of Alberta
- City = Edmonton
- City = Edmonton
- Province = Alberta
- Province = Alberta
- Country = Canada
- Country = Canada
- Postal Code = T5G 2G5
- Postal Code = T6G 0M4
- Telephone = 403-432-3762
- Telephone = 403-436-2628
- Address Type= Home
- Address Type= Business
- Social Insurance Number= 606 268 894
- Birth Date = Sept. 14, 1945

each other unless specifically linked by the file designer. This linking might occur when, for instance, a particular student had a particular staff member as an advisor. In this case, rather than enter data on the staff member into the student file, the file designer would set up a data element in the student file which would be called an advisor pointer. This element would contain only the identification of the staff record. If data were to be obtained from the student file, and the name and address of the advisor were needed, the advisor pointer would indicate which record in the staff file would contain these data. The program could immediately access the data from the staff file, then return to processing data from the student file. With this capability of linking separate subfiles in a file, no duplicate data need be kept.

2.5.2 Evolution and Facilities of SPIRES

A great deal of development has taken place since the original conception of SPIRES which was designed at and is currently supported by the Leland Stanford Junior University. Users of SPIRES have come to expect that the system will change almost daily. This would be a totally unacceptable situation in a commercial environment, as stability is one of the most important criteria for commercial systems. The development of SPIRES in an environment conducive to change has helped SPIRES stay as a "state of the art" DBMS.

For the university researcher, SPIRES also has one facility which is almost unheard of in commercial DBMS's. Each user is his own data base manager - that is each person is responsible for the maintenance of his own resources. For example, each user can allocate the amount of disk space to be used by each record or file. In a commercial system, a very senior systems person acts as a data base manager and assigns resources. Allowing each user to be his own data base manager gives those users tremendous freedom, again, something unacceptable in a commercial system. It also reduces human costs significantly as no professional data base manager must be hired.

2.5.3 Capabilities of SPIRES

The following are some capabilities which some authorities have suggested as extensions to existing DBMS's. The respective capabilities of SPIRES which satisfy these suggestions are discussed.

2.5.3.1 Numerical Manipulation

In order to use data stored in an information bank for forecasting, one must have the ability to easily manipulate that data, or to output it in a form that is easily input to a statistical analysis package. Picciano (1978, pp. 259-272) makes a case for having the DBMS output data in a format compatible with the Statistical Package for the Social Sciences (SPSS). His claim is that the conventional methods

of using specially written COBOL or FORTRAN programs which tabulate and analyse data output by DBMS are usually difficult to modify, while SPSS has predefined routines which perform many common statistical analyses.

Some Data Base Management Systems have a capability for numerical manipulation. This allows the writing of programs to analyse, tabulate and report results completely within the DBMS itself.

SPIRES has both these capabilities. It can output information directly into a SPSS compatible file, and it has numerical manipulation capabilities built into its protocol language. Hence, data stored in a form useful for obtaining data on individuals can quickly and easily be combined for reports on trends.

2.5.3.2 Privacy and Security

Users of information systems require control over the data which they are most competent to update or which they define as confidential. These are two major reasons often cited for requiring a separate data base for each application. Since the use of separate data bases may lead to the storage of redundant data, SPIRES has made provision for confidentiality and decentralized control of update ability within a single SPIRES file. As stated earlier, a SPIRES file is made up of a number of subfiles. Each subfile can be made accessible to every other subfile, but also can be used, maintained and controlled by an individual user. In

other words, the person who controls access to each subfile can specify who can examine or change each element in that subfile. This results in each application area being able to maintain its own subfiles, but if information is needed from some other subfile, it is immediately accessible if permitted by the person who controls access to the subfile. This can be thought of as an even more generalized application of the "Generalized File Synthesizer" of Richard and Richard(1978, p. 120).

2.6 Decision Support Systems

Observations about the inadequacies of the current uses of MIS's in educational administration were included earlier in this chapter. Recently, a new concept in computer use has emerged which synthesizes many of the criticisms of the current state of the use of computers in business, and offers a different approach to using computers in management decision making. This area is called decision support systems (DSS)[6].

Keen and Scott Morton give the following as their introduction to *Decision Support Systems: An Organizational Perspective* (1978):

> "Decision Support Systems (DSS) represent a point of view on the role of the computer in the management decisionmaking (sic) process. Decision support implies the use of computers to:

----

[6] "DSS: An Executive Mind-Support System" by Keen and Wagner (*Datamation*, Nov. 1979, pp. 117-122) provides a succinct statement of the philosophy of DSS.

1) Assist managers in their decision processes in semistructured tasks.
2) Support, rather than replace, managerial judgment.
3) Improve the effectiveness of decisionmaking rather than its efficiency." (1978, p. 1)

An examination of each these points, shows that the DSS philosophy offers an alternative to some of the current problems manifest in the use of computers.

2.6.1 Assist Managers in Their Decision Processes in Semistructured Tasks

Simon (1960, pp. 2-3) defines three stages within the problem solving process.

> "The first phase of the decisionmaking process - searching the environment for conditions calling for decision - I shall call intelligence activity (borrowing the military meaning of intelligence). The second phase - inventing, developing, and analyzing possible courses of action - I shall call design activity. The third phase - selecting a course of action from those available - I shall call choice activity....Generally speaking, intelligence activity precedes design, and design activity precedes choice. The cycle of phases is, however, far more complex than the sequence suggests. Each phase in making a particular decision is itself a complex decisionmaking process. The design phase, for example, may call for new intelligence activities; problems at any given level generate subproblems that in turn have their intelligence, design, and choice phases, and so on. There are wheels within wheels .... Nevertheless, the three large phases are often clearly discernible as the organizational decision process unfolds. They are closely related to the stages in problem solving first described by John Dewey: "What is the problem? What are the alternatives? Which is best?""

Keen and Scott Morton (1978, pp. 93-96) discuss three kinds of decisions:

1. Structured Decisions - decisions in which all three

phases (Intelligence, Design, and Choice) are structured. "Structured" means that one can specify algorithms or decision rules which allow the problem to be defined, alternative solutions to be specified, and a best solution to be selected.

2. Semi-structured Decisions - decisions in which one or two of the phases must be left completely in the manager's hands because one cannot define those stages precisely enough for a structure.
3. Unstructured Decisions - decisions in which one cannot define the conditions that allow recognition of the problem.

2.6.2 Support, Rather than Replace, Managerial Judgment

One of the major faults which many managers find with the field of Operations Research is that it allows some managers to transfer the total responsibility for making decisions from themselves to some numerical technique. In a totally structured decision this may be acceptable but many "technical" managers attempt to apply operations research techniques in semi or unstructured decisions.

Keen and Scott Morton (pp. 218) explain that:

> "A main reason for providing a DSS is to extend managers' "bounded rationality" in tasks that involve complexity of information and concepts. The DSS is partly a system for learning - better decision making should result from better understanding, richer insights, and more extensive assessment and synthesis of data."

2.6.3 Improve the Effectiveness of Decision Making Rather than its Efficiency

The difference between effectiveness and efficiency is one of the most important concepts that has consequence for the use of computers. Keen and Scott Morton (1978, p. 7) define the two concepts as follows:

> "Efficiency is performing a given task as well as possible in relation to some predefined performance criterion. Effectiveness involves identifying what should be done and ensuring that the chosen criterion is the relevant one."

As Clemson (1980, pp. 98-99) has identified there is a tendency for MIS systems personnel to cause the users of the system to suffer from information overload. Once data are stored in a computer, it is so simple to produce reports that data processing personnel can get carried away producing paper output. Instead, the data processing professional should be spending time finding out which particular piece of information is needed and providing it in an easy, quick manner.

In *Decision Support Systems: Current Practice and Continuing Challenges*, Steven Alter (1980) discusses the aim of DSS: improving the effectiveness of the manager. Alter(1980, pp. 95 - 108) identifies five ways in which DSS can accomplish this:

1. A DSS can improve personal efficiency by helping the manager do the same job in less time.
2. A DSS can expedite problem solving by:
   a. permitting fast turnaround - answers to simple

questions or the provision of data which will assist in decision making can be obtained in a matter of seconds rather than hours or days,

b. improving consistency and accuracy,

c. providing better ways of viewing problems which usually meant "there now existed access to information that had been previously either unavailable or available but in unusable form." (Alter, 1980, p. 99)

3. A DSS can facilitate interpersonal communication by providing individuals with tools of persuasion and by providing an organization-wide vocabulary and discipline.
4. A DSS can provide learning and training facilities which help users of the system understand the organization and the environment in which it operates.
5. A DSS can provide data for overall organizational control even though the main purpose is to provide data for individual decisions.

2.6.4 Summary of Decision Support Systems

Keen and Wagner (1979, p. 117) claim that a Decision Support System should be thought of as an "executive mind-support system."

A Decision Support System is a new view of the possible application of computer technology which should not only help managers make better decisions, but which should assist computer professionals.

"What about the remaining people who have been traditionally involved with computers and their applications ... One might think ... that the new easy-to-use development languages will leave them out entirely. But the fact is that DSS and the languages used can help these people make their products and services more complete. The technical specialists can work - almost for the first time from the user's view of the world." (Keen and Wagner, 1979, p. 122).

3. INSTRUCTIONAL DEPARTMENT COMPUTER SUPPORT SYSTEMS

In Chapter two many of the applications of computer technology which may be used in educational administration were reviewed. The areas of DBMS, MIS, and DSS can have direct applications for the middle level administrator. It was also stated that while system capabilities have been expanding, it has become possible for individual departments to investigate the possibility of developing computer support systems (CSS) which are designed for their specific and individual needs.

This chapter proposes a strategy for the design and implementation of an instructional department computer support system (IDCSS). The strategy is based upon assumptions that a department has little existing computer expertise, no existing computer support system and minimal financial resources for such an undertaking. A further assumption is that people in the department are aware of, and considering the implementation of an IDCSS as a possible alternative to the existing department information system. The strategy is written from the perspective of people promoting the examination of the use of an IDCSS in the department.

3.1 What is an Instructional Department Computer Support System?

There is no existing well accepted definition for an IDCSS. The following is offered as a definition:

> An instructional department computer support system is an integrated, computer based system which supports the day to day functioning as well as the long term planning of an instructional department.

An IDCSS may cover an extensive range of tasks from simple to complex. For example clerical tasks, such as the automatic generation of letters notifying each student of the name of his or her advisor and of the time of registration; to decision support material, such as the generation of staff workload reports used to aid in making course assignment decisions can be supported. Depending on the wishes of the members of the department, the IDCSS might include components such as:

1. a student record system,
2. a staff record system,
3. a course registration system,
4. a budgeting system, or
5. a department library system.

3.2 Attributes of an Instructional Department Computer Support System

While an IDCSS can have many different components and perform many different tasks, there are certain common attributes which an IDCSS should have. These attributes will allow for:

1. a good basic design of the IDCSS,
2. easy modification of the IDCSS,
3. easy expansion of the IDCSS,
4. the orderly introduction and installation of the IDCSS, and
5. an improved possibility for successful acceptance of the IDCSS by department members.

The proposed attributes of a good IDCSS can be divided into two groups: those which are oriented towards human acceptance and use of the system, and those attributes which are basically technical in nature. These two groups of attributes are highly interdependent, that is, it is not likely that a bad technical design will gain acceptance, while the best technical design which does not meet the perceived needs of the department members is equally unacceptable.

3.2.1 Human Oriented Attributes of an Instructional Department Computer Support System

> "In addition to providing for innovators and creating the conditions under which innovation thrives, we must also take care of the needs of the 'acceptors' - the majority of educators, those who must learn to accept and use the new resources. We must not be content with lamenting the fact that most people are heel-dragging resistors to change, suspicious of the new, and not very much interested in creating new things." (Caffrey, 1965, p. 14)

The need for the IDCSS must be perceived to come from within the department.

Members of the department must perceive that the existing information/support system is not meeting their needs, or that there is no current way to obtain information they need (Havelock, 1973, pp. 64-75).

The IDCSS should be perceived to evolve from the department's current information/support system.

People are ego involved with the existing information/support system. Havelock suggests that in designing a new system for a client, the best diagnostician

> "starts with the pain, the need as the client feels it, but he goes on to identify what is right with the client as well as what is wrong, and finally he puts these elements together to make a coherent picture of a total system which has goals and is striving to achieve those goals." (Havelock, 1973, p. 64).

The new system should appear to build on the strengths of the current system. If possible, initial reports produced by the IDCSS should follow the same format as previous reports to allow department members to perceive a smooth change over to the IDCSS.

The IDCSS must be amenable to change.

As people get used to the IDCSS, they will ask for modifications. The system must be able to respond quickly and relatively easily to these requests. The system design must anticipate and plan for change (Keen and Wagner, 1979, p. 118).

The IDCSS must be perceived as an aid to current practice and be supportive of as yet undefined needs.

It is imperative that potential users of the IDCSS perceive that the new system will not only assist current tasks, but that it has the capability to grow with, and indeed assist in the growth of, the department (Schwartz, 1970, p. 30).

The IDCSS must be easy to use.

The IDCSS must not make staff feel that they must be computer programmers in order to use the system (Keen and Wagner, 1979, p. 118).

The IDCSS must not be perceived as adding extra work without corresponding payoff.

The IDCSS must not be perceived as using resources which are better spent elsewhere.

3.2.2 Technical Attributes of an Instructional Department Computer Support System

The IDCSS must be open ended.

The IDCSS will, by the virtue of its existence, act as a change agent. Once people discover that certain tasks they thought impossible or too time consuming to do are performed virtually instantaneously by the IDCSS, they will start to ask for additions to the system (Keen and Wagner, 1979, p. 118).

The IDCSS must be easily modified.

Because the system will act as its own change agent, information which was at one time maintained in a form which was aggregated at a certain level may be requested at a more minute or more aggregated level (Keen and

Wagner, 1979, p. 118).

The IDCSS must be integrated.

It sometimes appears easier to develop systems individually; for instance a budgeting system, a staff record system, a library system, and a student record system could all be developed independently. While initially it may seem like less effort to develop independent systems, in the long run this is false economy. Once the 'data are in the computer', people will start to see the relationships among the data and will start to ask for combinations of information which span individual systems. If these systems are independent, it will be virtually impossible to combine information. If, on the other hand, these systems are really subsystems in an integrated IDCSS, then combining data to provide the requested information becomes possible.

The IDCSS should be designed under a Data Base Management System.

As the system evolves it may have a completely different appearance to one type of user, an administrator, than to a second type of user, a clerk. The administrator may perceive the IDCSS as a way to access information to answer questions. The output from the IDCSS might appear to this administrator as unformatted responses to questions. For example, the administrator might ask the question "How many full time M.Ed. students are enrolled

in this department?" (worded differently so that the computer understood the question). The computer might respond "76 student(s)". Clerks, on the other hand, may perceive the IDCSS as a way to reduce many of the repetitive tasks of their job. For example, the IDCSS might appear to them as a way of providing a formatted list of names and addresses for currently enrolled full time M.Ed. students without the necessity of retyping the list each time a student enrolls or convocates.

The IDCSS thus would show two different "faces" to these two groups of users. The clerks may in fact not realize that they are using the same system as are the administrators. The underlying similarity is that the same data are operated upon. In order to easily and economically facilitate these different appearances, the system should be built under a good DBMS. The use of a DBMS will also facilitate unanticipated uses of the data (Senda, 1977, p. 97), that is once the data have been placed in a DBMS, they may be combined in different ways to answer questions which had not been anticipated at the time the IDCSS was specified.

3.3 A Strategy for the Design and Implementation of an Instructional Department Computer Support System

In the following discussion, it is assumed that a department currently operates with a manual support system. That is, files are maintained manually and information from

them is obtained and cumulated manually. An IDCSS is being considered as a replacement for the current system.

The basic strategy for the design and implementation of an IDCSS is based on theories of how systems adopt and adapt to change[7]. An IDCSS may cause a special kind of series of changes. While the implementation of an IDCSS may originally be intended to ease clerical tasks associated with the operation of a department, it can soon affect the kind of data, the organization of the data, and the accessibility of the data maintained by the department. A change in the kind, organization or accessibility of data in an organization can lead to organizational changes. This leads to possible modification of the assumptions which underlie the structure of the instructional department computer support system itself (Schwartz, 1970, p. 30), which in turn leads to necessary modifications of the IDCSS. The system designer must constantly be evaluating the IDCSS and must be willing to redesign, or even redefine the system at any time. In order to facilitate continuous change, the feedback cycles in the change process must be very fast, that is, as well as changes to the system being made as quickly as possible after a request, users of the system must be made aware that changes were made as a result of their request as soon as

----

[7] Havelock (1973) provides a guide to change and innovation in Education. Zand and Sorenson (1975, pp. 532-545) provide a theory of change in Management Science based upon the Lewin-Schein theory of change. The Zand and Sorenson theory has been used extensively as the change theory model by the researchers in Decision Support Systems (Keen and Scott Morton, 1978; Keen and Wagner, 1979; and Alter, 1980).

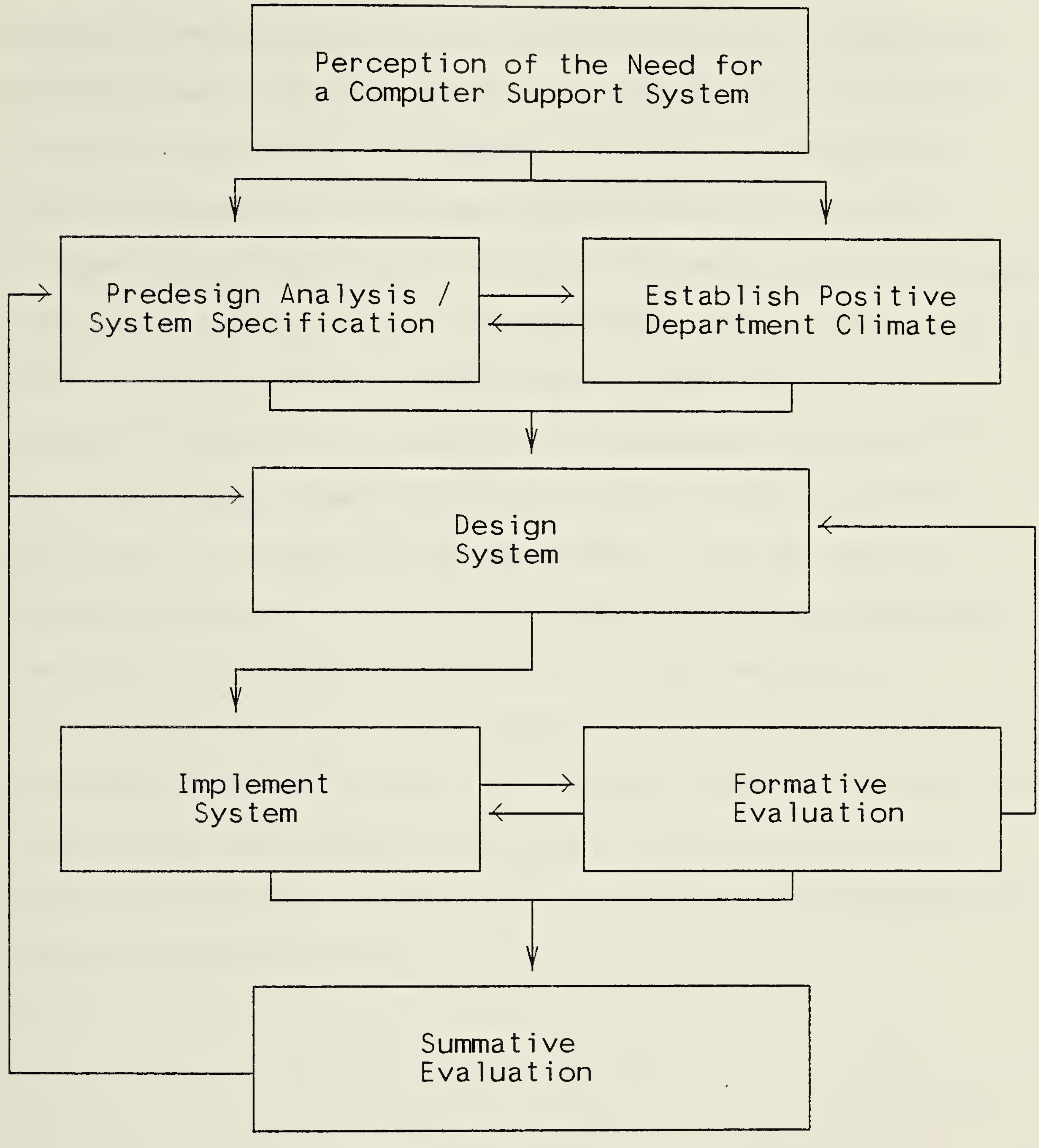
possible.

Figure 3.1 shows the general strategy which the author has developed for the design and implementation of an IDCSS. The strategy consists of seven stages, some of which operate in parallel with each other. As stated at the beginning of this chapter, the strategy is written from the perspective of someone in the department who is aware of the need for a change to a current manual information/support system, and who wishes to assess the possibility of using an IDCSS to replace the current manual system. The first three stages in the strategy are independent of the choice to implement an IDCSS, but they are written from the point of view of a person proposing an IDCSS as the preferred alternative.

3.3.1 Stage 1 - Perception of the Need for a Computer Support System

The first step in the development of an instructional department computer support system is the realization by members of the organization that there is a need for such a system. This realization usually is negatively indicated, that is it becomes more and more apparent that the current system is no longer meeting the needs of the department either effectively or efficiently as these terms were defined in Section 2.6.3. In examining how to improve the existing system, one alternative may be the implementation of a completely new system, possibly an IDCSS.

Figure 3.1 A Strategy for the Design and Implementation of an Instructional Department Computer Support System



Havelock (1973, pp. 119-121) discusses the problem that the perception of the need for a change in the current system is not going to be shared equally by all members of the department. Even those who see the need for change will not agree as to what the change should be. During this stage, proponents of the development of an IDCSS must arrange the introduction of this topic very carefully. Great care must be taken to assure that those who are skeptical of such a solution do not become active opponents merely because a computer is involved in the suggested solution[8]. There are enough legitimate reasons not to use an IDCSS (e.g. cost, availability of equipment, availability of trained personnel, etc.) without having to introduce some of the irrational reasons cited by Oettinger and Marks.

Proponents of an IDCSS should at this point attempt to gain acceptance of the need for change, and that one of the alternatives for change should be an IDCSS. The Lewin-Schein[9] theory would refer to this as the beginning of the unfreezing process.

----

[8] The book *Run, Computer, Run: The Mythology of Educational Innovation* by Oettinger and Marks provides case studies and analysis of the rejection of technological innovations, and particularly the introduction of computers in education.
[9] Lewin (1952) and Schein (1961) established a framework of organizational change based on the stages of unfreezing, moving and refreezing. Zand and Sorenson (1975, pp. 532-545) provides a study of how favorable and unfavorable forces on each of the stages affects the acceptance or rejection of the change.

3.3.2 Stage 2 - Predesign Analysis / System Specification and Stage 3 - Establish Positive Department Climate

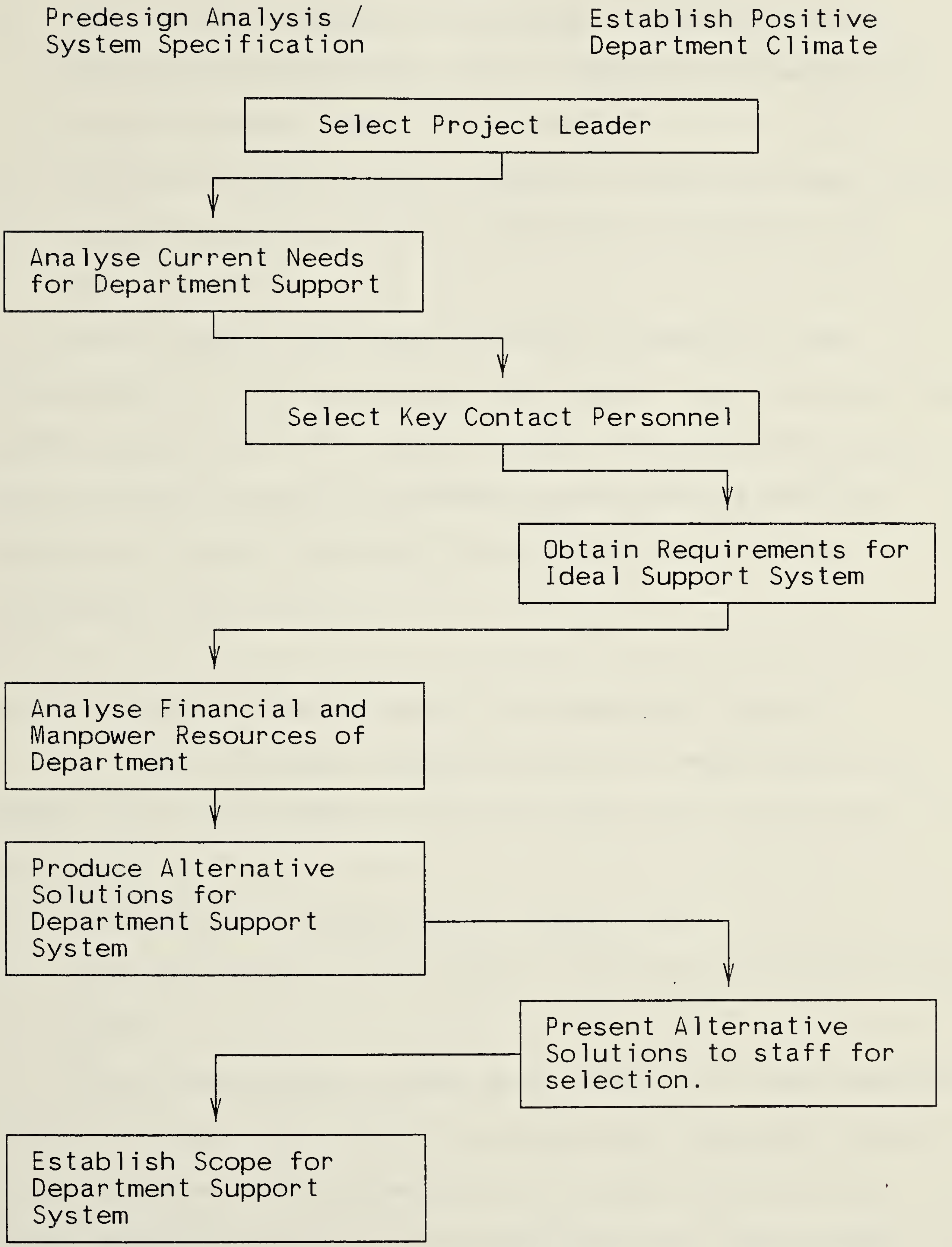
Once there is acceptance of the need for change, the process moves into two parallel and highly interdependent stages: the Predesign Analysis / System Specification Stage and the Establish Positive Department Climate Stage. Due to the high degree of interrelationship of activities occurring during these two stages, tasks can only be differentiated along fairly artificial lines. Those tasks which are essentially analytic or technical in nature have been placed in the Predesign Analysis / System Specification Stage. Those tasks which tend towards the acceptance and assimilation of the system are placed in the Establish Positive Department Climate Stage. Certain tasks such as the selection of Key Project Personnel may fit into both Stages.

While it is impossible to suggest a perfect order in which to undertake tasks in these two stages, Figure 3.2 gives a recommended order. An explanation of each task follows.

3.3.2.1 Selection of a Project Leader

The first and perhaps most important step is the selection of a Project Leader. In many applications, because of limited resources, this person may be administrator, designer, implementer and evaluator of the project, therefore he or she must possess varied qualifications. The Project Leader must be:

Figure 3.2 Tasks in the Predesign Analysis / System Specification and Establish Positive Department Climate Stages



1. familiar with the kind of organization in which the system is to be implemented,
2. familiar with computer technology - especially those particular applications such as DBMS and MIS which may be particularly useful in an IDCSS,
3. non-threatening to staff members in the department, and
4. acceptable as an expert in the field of IDCSS by the staff members.

The Project Leader may be either a member of the department or from outside the department. He or she must be given a great deal of freedom inside the organization and should report directly to someone with both official and unofficial power, preferably the Department Chairman (Havelock, 1973, p. 53).

3.3.2.2 Analyse Current Needs for Department Support

In order to begin diagnosing what is required for an IDCSS, one first needs to establish what the current DSS is doing by:

1. examining the current paper system, and
2. interviewing the staff members involved with the current system.

The current paper system gives examples of the kinds of forms, letters, etc., which the department receives from or sends to others. The summation of the data collected on these forms and letters is the data base for the current paper system. In order to assess what data are being

collected, one must examine different types of files (e.g. student files, staff files), university forms, etc.

When interviewing staff, both academic and support staff members from all types of job positions should be queried so that tasks and the interrelationship of tasks in the department can be assessed. These staff members can also identify what data are needed to perform these tasks.

The Project Leader should analyse the forms and the interview responses in order to establish a comprehensive list of data elements which are currently required. Notations of where each data element is to be obtained should be prepared. A cross-indexing of relationships between data elements should also be made. Finally, notation should be made of where each data element is required to produce existing reports or to support the decision making process.

3.3.2.3 Select Key Contact Personnel

Members of the department primarily involved in the administration of the department will have a strong interest in the development of any IDCSS. Other members of the department, primarily instructional and support staff, may hold strong feelings about any proposed IDCSS even though it may not have a strong direct effect on their jobs. Both of these groups may include:

1. individuals who are very interested in a new system, and who would like to be actively involved in the

development of the IDCSS,

2. individuals who support the implementation of a new system but who do not want to be actively involved in the development, and
3. those who do not support the new system and might actively campaign against implementation.

The Project Leader, in consultation with the Department Chairman, should select a few individuals from each of these groups who can be used as "sounding boards" on whom to test proposed ideas for the new system.

Rather than a formally designated committee this should be an informal group of department staff whose inclusion will greatly improve the specifications of the capabilities of the system, and/or, because of their informal power in the organization, may help to pre-empt any concerted opposition to the system.

3.3.2.4 Obtain Requirements for Ideal Support System

Members of the department have worked with the existing information/support system and can identify both good and poor characteristics. In order to improve the possibility of an IDCSS being accepted, members of the department must feel that they were involved in specifying what the new system *should* and *should not* do.

For analysis purposes, the members of an instructional department can be arbitrarily divided into four functional groups. Most individuals will fall into one group only, but

some may fall into more than one group. The four groups, and a description of tasks within each group that may affect the structure of an IDCSS are listed below:

1. Administrative staff
   a. make decisions in areas such as the admissibility of students,
   b. monitor a student's program,
   c. assign staff advisors to students,
   d. budget for capital and operating expenses in the department, and
   e. make staff tenure and promotion recommendations.
2. Faculty and department planners
   a. plan future course offerings,
   b. plan program alternatives,
   c. plan class sizes,
   d. decide upon staff hirings and replacements, and
   e. budget for long term capital expenditure and replacements.
3. Academic staff may need to
   a. access individual student files for purposes such as program counselling, or
   b. search a library file.
4. Secretarial staff will be constantly using the IDCSS as they update data and access information requested of them by other staff.

As mentioned earlier, each of these groups have different needs and perceive an IDCSS differently. Members of each group should be asked to give their impressions of what the ideal IDCSS should include.

In considering an IDCSS, the Project Leader must avoid two common traps which have weakened the design of many past computer systems.

1. The author's experience as a computer consultant for approximately ten years, indicates that many clients are extremely naive about the capabilities of computers. One of the most common problems encountered with clients is to have them explain what they would like done. Instead, many insist on explaining what they think the computer can do for them. Unless the consultant can convince the client to drop any preconceived notions of what a computer does, such clients quite often end up with a system which is significantly inferior to what could have been developed. Personal discussions with other computer consultants in similar positions (Senda 1975-80, Davis 1975-80, James 1975-80, for example) confirm this observation.
2. Conversely, a great part of the business of many computer consultants is selling "package" systems (Hussain, 1973, p. 313). A major problem may arise when a consultant tries to fit a new application into an existing program package. This type of approach can lead to extremely bad feelings on the part of the customer,

and a poor reputation for that consultant if the package fails to meet the expectations of the customer (Scheinbien, 1980, class presentation).

The Project Leader must emphasize to the members of the department that they are being asked to provide what they would like to see in the ideal IDCSS, not what they think is either feasible or within the budget of the department. The Project Leader must also keep an open, receptive mind so that he records what the various staff members report as desirable, not what fits in with an existing system with which the Project Leader is familiar, or a system he would like to market.

Specifications for the ideal system may be obtained in a number of ways:

1. Group meetings can be held with members of each group, explaining what is being done. The attempt in this meeting should be to establish a "brainstorming" atmosphere so that people get ideas from and amplify each other's thoughts and concepts.
2. Individuals in the different groups should be asked to keep a diary for a specified period of time, during which they note down the kinds of information they need, how these needs are or are not being met by the current system, and any ideas they have about improved or new capabilities they would like to see.
3. The Project Leader should meet informally with individuals from the department in order to elicit ideas

which would not have been presented in large group meetings.

4. There are a number of people performing the functions of administrative staff and faculty and department planners in a faculty. They include the Dean, Associate and Assistant Deans, Chairmen, Administrative Professional Officers, and a few special appointments. In order to obtain as many different ideas as possible, a representative sample of these individuals should be interviewed as they may have different perceptions of what could be designed into the new IDCSS to be of use to a person in this kind of activity.

3.3.2.5 Analyse Financial and Manpower Resources of Department

The financial and manpower resources which can be dedicated to the IDCSS may be limited. The majority of the direct costs in the design and management of the project will be incurred by the Project Leader. Hence, once the Project Leader is hired or assigned, much of the fixed manpower expenses will be known.

In the special case of the design and implementation of an IDCSS, the following additional resources will be needed:

1. clerical manpower - for entry of data to the computer system,
2. computer processing costs, and
3. computer terminal purchase/rental costs.

During this analysis, it should be decided if it is possible to reassign manpower and equipment from existing department resources. For example, if clerical staff have periods of time when their workload is low, this time may be used for entering archival information. Certain costs associated with an IDCSS may not become actual "hard money" charges to department budgets, but may be services generally available to all departments at no charge or at some fixed charge regardless of amount of use. Computer terminals may already exist in the department for use in data analysis or in text processing.

3.3.2.6 Produce Alternative Solutions for Department Support System

Once it is known what the current need for an IDCSS is, what the collective staff requirements for an ideal IDCSS would be, and what the department resources are, the Project Leader can produce a set of alternative solutions for an IDCSS. These solutions may range from staying with the status quo, to implementing a very sophisticated computerized IDCSS.

In a proposal for an IDCSS, the state of available computer facilities must be investigated. It is one thing to consider implementing an IDCSS under a very sophisticated and powerful DBMS (such as SPIRES, discussed in Chapter 2), but it is completely another to consider implementing the

same system without using a DBMS approach[10]. With each proposed solution, the Project Leader should provide an estimated budget and time line for implementation.

The Project Leader should then consult with the Department Chairman and the Key Contact Personnel to get feedback on each of the proposed systems. The Department Chairman and the Project Leader should select the alternative which they feel is best for the department, and a final proposal should be prepared for this alternative.

3.3.2.7 Present Alternative Solutions to Staff for Selection

The alternative solutions should be presented to the members of the department. At this time it must be made clear to the department members that any system proposed at this time is only the first approximation to a final solution. The *evolutionary* nature of an IDCSS must be

----

[10] Ron Senda is the Group Leader for the Information Managment Group at the University of Alberta's Department of Computing Services. As such, one of his tasks is to estimate manpower and computer costs for the implementation of MIS and DBMS systems. In an invited presentation to a class in the course entitled *Computing Concepts in Educational Administration* on August 7, 1980, Mr. Senda made the following statement:

> "I have a standard statement which I usually tell prospective clients who are considering using either SPIRES or a conventional program designed exclusively for their data base management task. If it would take a day to program the task conventionally, we can do it in SPIRES in an hour; if it would take a week to program the task conventionally, we can do it in SPIRES in a day; if it would take a month to program the task conventionally, we can do it in SPIRES in a week; and if it would take a year to program the task conventionally, we can do it in SPIRES in a month."

emphasised, and it should be explained that the department members should feel free to bring forward requests or suggestions as the project progresses. The solution suggested by the Department Chairman and the Project Leader should be indicated and the reasons for its choice stated. Members of the department should be able to specify modifications to this proposal. If the IDCSS chosen is substantially changed by the suggestions from the staff, the Project Leader should design a new pilot proposal to meet these new specifications, prepare a new budget and present this to the staff once again for acceptance.

Since this thesis is involved with the development of an IDCSS, from this point on the model assumes that an IDCSS was chosen to be implemented as the new Department Support System. This IDCSS will be referred to as the prototype IDCSS.

3.3.2.8 Establish Scope for the Instructional Department Computer Support System

Once department consensus has been reached on the general guidelines for the IDCSS, the Project Leader and the Department Chairman should draw up a set of boundary conditions. These define the areas which the IDCSS must include, those areas into which it may move at the discretion of the Project Leader, and those areas into which it may not move. The boundary conditions may include:

1. the nature of the information to be stored by the

system,

2. cost,
3. use of department manpower resources,
4. reports to be generated by the IDCSS,
5. security of the IDCSS, or
6. availability of information from the IDCSS.

The boundary conditions for the prototype IDCSS can be indicated as shown in Figure 3.3.

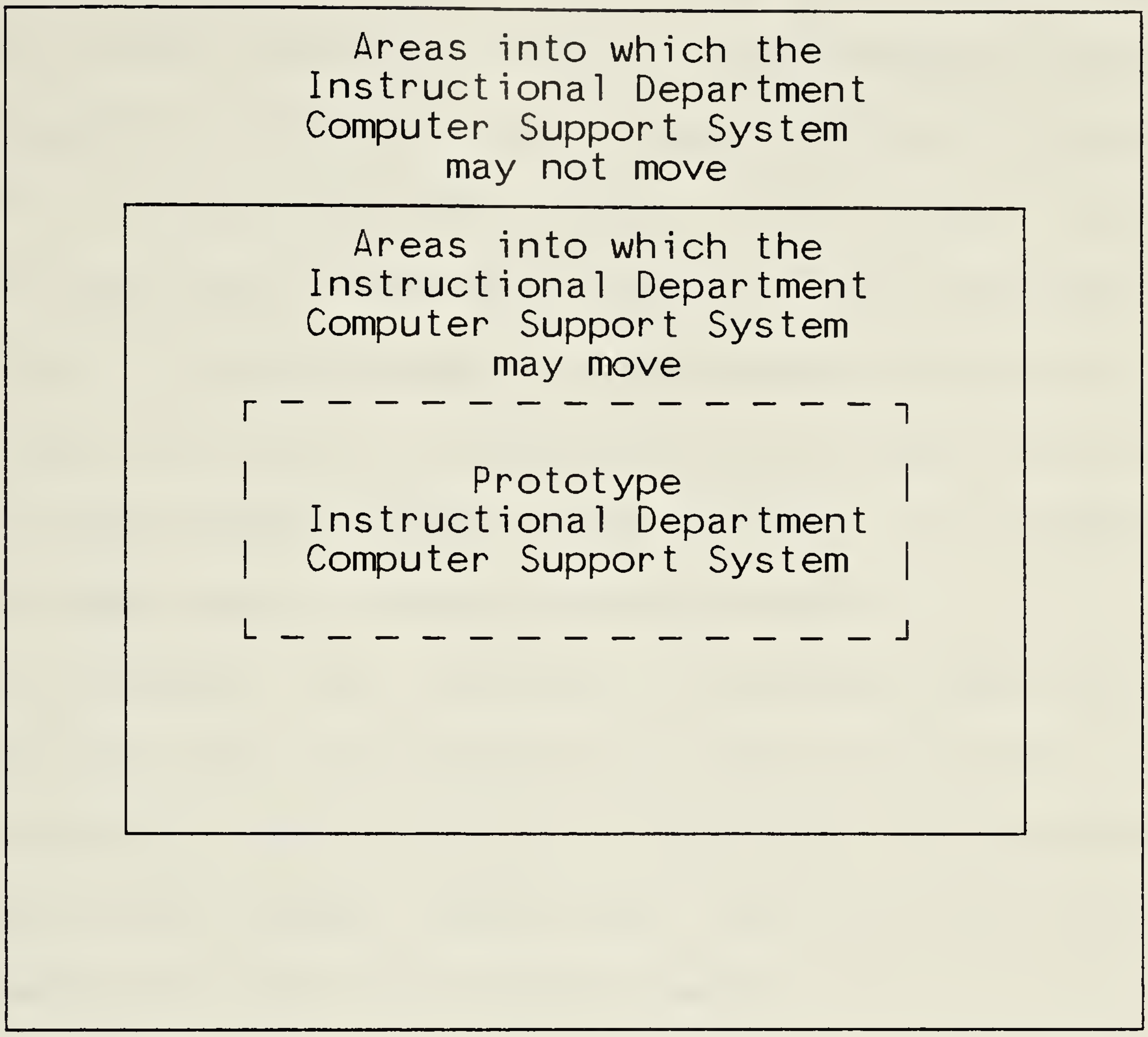
3.3.3 Stage 4 - Design System

The specification and design of the IDCSS should be relatively independent of the physical machine or DBMS upon which it is to be implemented. In contrast, the system implementation is almost completely dependent upon the DBMS used. While the System Designer and/or the Project Leader must be aware of the general capabilities and constraints of the DBMS upon which the IDCSS will be implemented, the main intent during this stage should be to design a system which will stand on its own merits and which could, if necessary, be implemented on a DBMS different than that which was originally intended.

During this stage, the System Designer must address the following general questions:

1. What elements will be stored in the data base?
2. What will each element of data look like?
3. How do elements relate to each other?
4. How will data be entered into the data base?

Figure 3.3 Scope of an Instructional Department Computer Support System



In this figure, lines indicate boundaries between possible areas within the scope of the IDCSS. The area inside the dotted lines indicate tasks which the IDCSS must include - this is referred to as the Prototype System.

If a task is requested which requires movement from within the Prototype System across a dotted line, then this task can be undertaken if resources exist.

If a task is requested which would require crossing a solid line, then this task must not be undertaken.

5. How will information be produced from the data base?

The System Design Stage is a detailing of the prototype system. In the Predesign Analysis / System Specification Stages general statements about data to be maintained and reports to be generated were made. In this stage everything must be described at the most specific level possible.

The five questions above generate more specific questions which must be answered for each data element.

1. **If a particular element is to be kept, is it required for each record, or may it be optional?**

   In a student file, the Social Insurance Number may be required, but the maiden name might be optional.

2. **Is there a limited number of possible values which a particular element might take, and if so, can this element be codified to save space?**

   One element in a file might designate a department. In the Faculty of Education, for example, there are five departments. Each student and staff member is resident in one of these five departments. If these departments were coded as 1, 2, 3, 4, and 5, rather than "Department of Educational Administration", "Department of Vocational and Industrial Education", etc. the amount of space in each record could be reduced by a significant amount (thirty nine characters in the case of the Department of Educational Administration). Over a file of significant size, this can be a considerable saving.

3. **Might the same element occur more than one time?**

The Social Insurance Number should occur only once in a student record, while that student may have many occurrences of course marks.

4. **Are elements of a fixed length or number of characters, or can the number of characters vary?**

   The Social Insurance Number has nine characters, while the number of characters in a person's name can vary.

5. **Who will have the ability and right to enter, modify and examine a particular element?**

   An instructor might be given the right to enter a mark for a student. Modification of the mark might be restricted to an administrative officer. Any academic staff member in the department might have the right to view the mark.

6. **Must particular elements be grouped together in a structure?**

   It is possible that a student might have three addresses, consisting of the data elements street, city, province, and postal code. Each address must be grouped together in a structure, otherwise there would be three streets, three cities, three provinces and three postal codes; but no knowledge of which street referred to which city, province and postal code.

Most importantly, there must be some method for merging all the data elements, and their relationships into some kind of a coherent system. A widely accepted approach to system design is "top down design" (Antworth, 1980, p. 182).

One of the more rigorous approaches which uses the top down design is "structured design", defined by Yourdon and Constantine (1975, p.7) as:

> " the art of designing the components of a system and the interrelationship between those components in the best possible way."

The top down approach to structured design would look at the design of an IDCSS in the following way.

1. Define the major objectives for the IDCSS.
2. The major kinds of information should be identified and grouped together.
3. Sub-groups of similar information within each group should be identified.
4. Sub-sub-groups of similar information within each sub-group should be identified.

This top-down approach is continued until the data element level is reached.

To demonstrate this approach, examine an IDCSS in which information on students and staff members is to be maintained. At the top level, this would be viewed as a single system. Immediately below that, the student information and the staff information would be separated. The top down process would then examine these two sub-systems (student information and staff information) independently. The student record would be divided into personal information, program information, and mark information. In further examining the program information, it might be necessary to have information on the student's

advisor, but here rather than storing information on the advisor, the system would point back to a record in the staff file. Figure 3.4 shows this top down approach in a pictorial format.

Once the system has been designed, the Project Leader must document the design of the system so that the system can be implemented. This can be accomplished by building a "data element dictionary" (Hussain, 1973, pp.165-166). A data element dictionary follows the structure of the system imposed by the System Designer. Each structure and data element is totally defined starting from the top of each file. Examples of entries of a structure and an element in a data element dictionary are shown in Figures 3.5 and 3.6 respectively.

During this stage, methods and forms for the collection and entry of data should be designed. Samples of proposed reports showing their format and proposed content should also be designed.

3.3.4 Stage 5 - Implement System

Since the IDCSS must be programmed on a particular DBMS, the actual implementation of the system will be almost completely dependent upon the programming language of that DBMS. Some DBMS systems (such as SPIRES) suit themselves very well to implementing a system which can evolve. Some other DBMS systems are not designed to allow the data structure to change once the original data structure has

Figure 3.4 Top Down Design For a Staff / Student Instructional Department Computer Support System

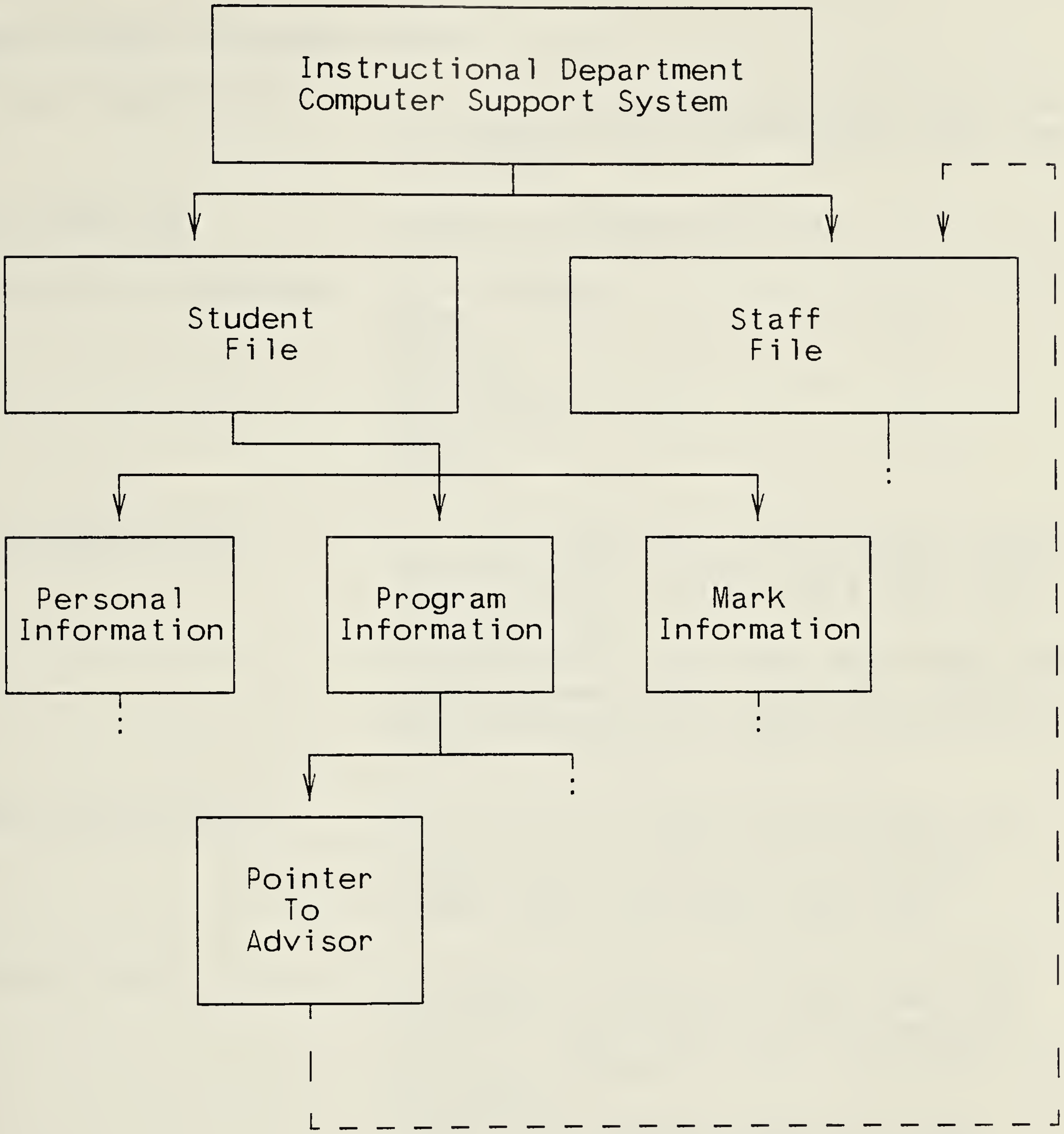


Figure 3.5 Data Element Dictionary - Sample Data Structure Specification

Name of Data Structure:	Marks
Literal Description:	The marks structure contains all the information on a particular mark for a student.
File Location:	Student Information File
Elements in Structure:	Course Name Year Session Section Instructor Mark Ruling
Data Description:	All the data on an individual course are entered in this structure. This structure will occur more than once in the file. This structure will be searchable on Course Name, Year, Session, Section, and Instructor in order to generate lists of students enrolled.
Level of Accuracy:	The level of accuracy must be very high, since this is the official record of the courses a student takes, and the marks obtained.
Access level:	Will be originally defined by the Chairman's office at the time the student registers for the course. Can be accessed by the Instructor or Chairman. Cannot be changed except by the Instructor who entered the mark.
Life cycle:	Entered at registration time. The actual mark and ruling can only be entered or updated by the Instructor. Purged only when the entire record is purged.

Figure 3.6 Data Element Dictionary - Sample Data Element Specification

Name of Data Element:	Course Name
Literal Description:	The official Course name as defined in the University Calendar.
File Location:	Student Information File
Data Description:	The course name will be precisely as defined in the University Calendar: e.g. EDADM511, or or EDPSY502.
Level of Accuracy:	The level of accuracy must be very high, since this is the official record of the courses a student takes.
Access level:	Will be originally defined by the Chairman's office at the time the student registers for the course. Can be accessed by the Instructor or Chairman. Can only be changed by the Chairman's office, using the "Error in Registration" procedure.
Life cycle:	Entered at registration time. Can only be changed by the Chairman's office, using the "Error in Registration" procedure. Purged only when the entire record is purged.

been defined[11]. The Project Leader will have to insist that the programmer implementing an IDCSS on a system of the latter type makes provision to allow for the system to evolve.

Figure 3.7 gives a flow diagram for the Implementation Stage. The intention of this flowchart is to show an order for implementing the system *as originally specified*. It is quite likely that as prototype reports are generated, information will be requested by department members which has not been included in the original prototype system. The Project Leader must decide whether the IDCSS should be modified at this time to include the missing data, or if modifications should wait for the first formal evaluation once implementation is complete. Senda (1977, p. 97) suggests that if the change requested is within the scope of the project and will not result in a major time delay to the project, minor revisions should be accommodated as the project is implemented.

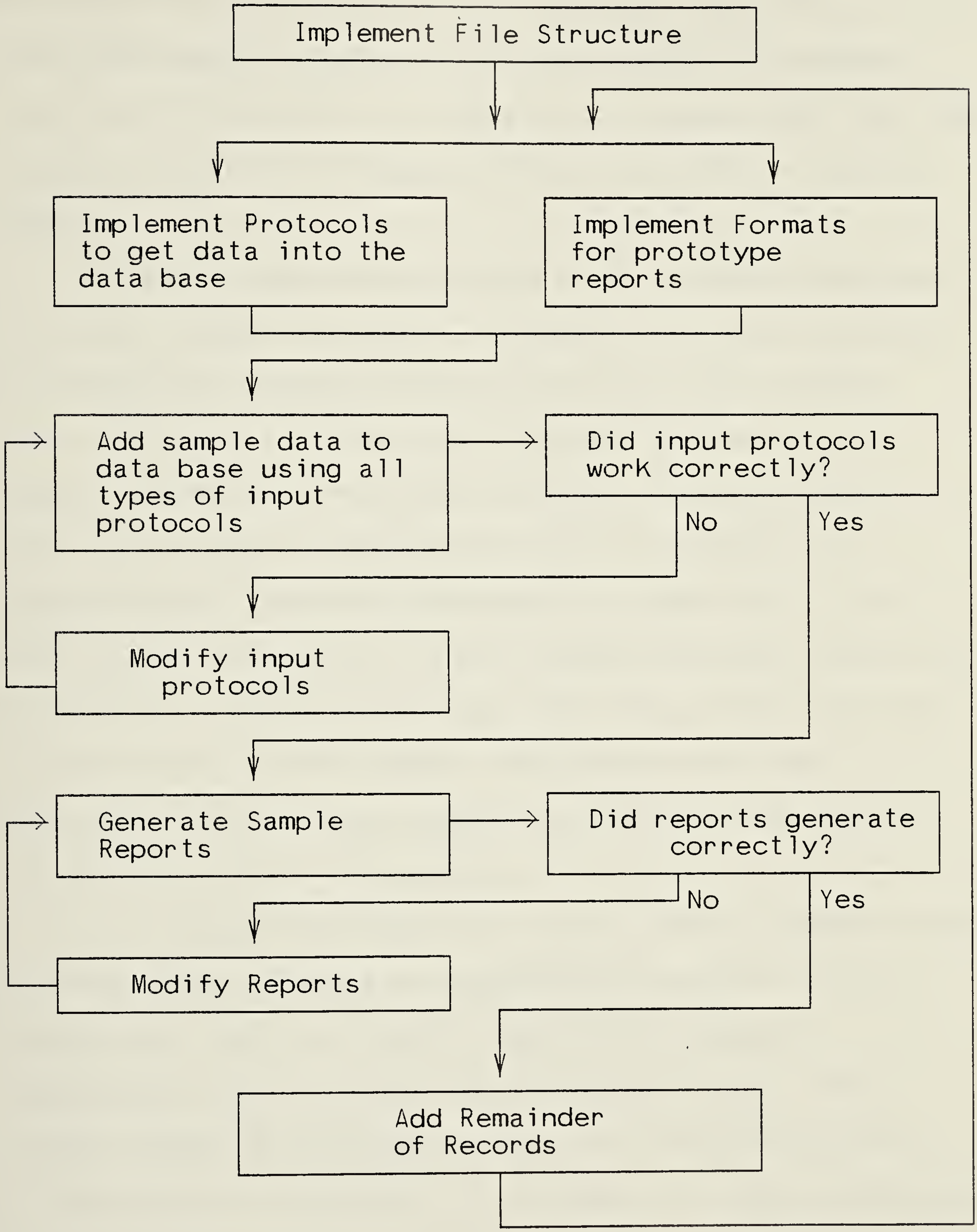
3.3.5 Stage 6 - Formative Evaluation

Throughout this discussion of an IDCSS, it has been stressed that the development is an evolutionary process. The formative evaluation stage emphasises this assertion. The concept of formative evaluation has been adapted from Scriven's paper on curriculum evaluation (Scriven, 1967; pp.

----

[11] See, for example, the micro computer based DBMS Selector III (Micro-Ap, 1978).

Figure 3.7 The Implement System Stage



39-83). Rephrasing Scriven, formative evaluation can be defined as constant evaluation: field testing work as it is being developed, and then using the feedback to produce revisions in the work. In Figure 3.1 it can be seen that the Formative Evaluation Stage parallels the Implement System Stage.

Formative evaluation as applied to an IDCSS development is a very informal process. It consists of using the IDCSS to the maximum extent possible at the earliest possible time. The results of this use (reports, statistics, or simply information) should be given to the appropriate staff. The Project Leader should ask for comments or suggestions for further improvements or additions. Those staff identified earlier as Key Contact Personnel should be particularly helpful since they have been closely involved in the project to this point. Other members of the department may have good suggestions because they have a less biased view of the system.

It is imperative that the Project Leader respond to and act upon these requests as promptly as possible. It is recommended that the Project Leader never respond immediately that a certain request cannot be met. Each request should be examined to see if it falls within the defined scope of the IDCSS. If it does not, this explanation should be passed back to the person making the request.. If the request is within the scope of the project, and the fulfillment of the request is a simple change or addition to

the implementation, then this requested revision should be made.

Some requests will be within the scope of the project, but will not be able to be met within the current implementation. This may mean partial redesign of the system, or a major rebuilding of the complete IDCSS. After conferring with the department member requesting the change as to the reasons that the redesigned system would be better than the existing system, the Project Leader and the Department Chairman, in conjunction with appropriate key contact personnel, will have to decide if the change should be made. Referring again to Figure 3.1, this may mean looping back to the Design System Stage.

The cost of design changes can be extensive, depending on the DBMS under which the IDCSS is being implemented and the number of records in the system. Some DBMS's, such as SPIRES, can accept major data restructuring or even major design changes with minimum computer cost and human time. As stated earlier a small design change in some other systems, such as the addition of one element of data, may involve rebuilding of the complete system and re-entry of all the data records.

Since the object of the IDCSS is to support the needs of the department members, if a department member determines that a system change is necessary and the Project Leader can ensure that this change will not be detrimental to the overall system, the change should be implemented.

The Project Leader should run the Formative Evaluation Stage in parallel with the System Implementation Stage. As new features are added to the system, the results of these new features should be shown to the staff for their comments and suggestions. It has been observed that the faster the Project Leader can react to suggestions, either to effect change, or to explain why the request cannot be met, the more willing staff members become to discuss the system and suggest further enhancements.

3.3.6 Stage 7 - Summative Evaluation

The concept of summative evaluation is also adapted from Scriven. Summative evaluation will be defined here as a terminal, overall, or outcome evaluation which indicates whether the method or approach being evaluated has met the stated objectives (based on Scriven, 1967, pp. 39-83). One of the hardest decisions to make when implementing a system which is designed to be constantly evolving is determining when the system is stable and a summative evaluation can be undertaken. Indicators which may signify that a summative evaluation might be timely include:

1. Requests for changes have become infrequent and minor (that is, the system seems to have reached a natural stabilization).
2. A number of requests are for information or services which are outside the defined scope of the current IDCSS. This may mean that the IDCSS needs to be

expanded.

3. Some external influence forces a summative evaluation when the system is still evolving. This could be a formal request for an evaluation from some outside funding agency. This type of untimely forced evaluation could cause premature rejection of a project which was still developing towards meeting the original objectives.

As stated, the purpose of the summative evaluation is to find out if and how well the IDCSS has met the original objectives, and to see in what ways it has departed from those original objectives. Possible outcomes of the Summative Evaluation include:

1. acceptance of the IDCSS,
2. rejection of the IDCSS,
3. establishment of the need for major modifications to the IDCSS, or
4. a decision to go back to the Predesign Analysis Stage to specify a new system.

The latter two alternatives may be necessary if the needs of the department have undergone major changes. These changes may, in part, be due to the implementation of the IDCSS.

4. DESIGN AND IMPLEMENTATION OF THE DEPARTMENT OF EDUCATIONAL ADMINISTRATION COMPUTER SUPPORT SYSTEM

Chapter three developed a strategy for the implementation of an IDCSS. This strategy is the result of a synthesis of many theories from different authors. The operational outcome of the strategy was a set of guidelines which should result in the successful implementation of an IDCSS.

In order to test the strategy, the Department of Educational Administration at the University of Alberta consented to undertake an IDCSS. The project attempted to follow the strategy set out in Chapter three as closely as possible. In this chapter, the actual process which led to the creation of the Department of Educational Administration Computer Support System (DEACSS) will be described in a manner parallel to the description of the strategy as described in Chapter three.

4.1 Stage 1 - Perception of Need for a Computer Support System

On July 1, 1977, a new Chairman of the Department of Educational Administration at the University of Alberta was appointed. The new Chairman had previously taught courses in computer applications to educational administrators and was aware of the possible applications of computer technology to assist in both the repetitive clerical tasks and in the decision making process within the department. The Chairman

felt that one of the projects which should be undertaken during his tenure in office should be the automation and provision of faster access to graduate student information.

In 1975, the University of Alberta had decided to implement the SPIRES system as a DBMS for researchers on the campus. In 1977 the author spent one week at Stanford University at a conference of SPIRES users discussing the capabilities of the SPIRES system. From these discussions, the author concluded that the design of SPIRES might facilitate the implementation of a graduate student record system.

In early 1978 the author proposed to the Chairman of the Department of Educational Administration that the facilities were now available on which a graduate student record system for the department could be built. The author suggested that, if the department was willing to participate and would provide clerical assistance and computing costs, the author would undertake a research project to design and implement a graduate student record system for the department. The Chairman took this proposal to the department members, and in late 1978 the department agreed to participate.

4.2 Stage 2 - Predesign Analysis / System Specification and Stage 3 - Establish Positive Department Climate

The department's commitment of resources to the project

was to provide computing costs and whatever time the Department Secretary could give to the project without seriously affecting her normal duties. The Department Secretary was to act as the main person in charge of collecting, collating, entering and correcting information in the IDCSS. The Department Secretary also became the main counsel on clerical needs of the department. The author became the Project Leader and the Department Chairman agreed to provide counsel on the administrative needs of the Department.

The Department Secretary provided the Project Leader with a "sample graduate student file" which contained copies of all possible forms which the department receives or sends out in conjunction with graduate students. The Project Leader interviewed department staff: the Chairman, Department Secretary, Administrative Assistant, and academic staff members, regarding their perceptions about the requirements of a department graduate student record system.

The Project Leader then analysed the information obtained from both the "sample graduate student file" and from the interviews in order to produce a set of initial specifications for the graduate student record system. Through this analysis it became apparent that graduate student records impact and are impacted by a great many other areas of the operation of the department. Staff members serve on graduate student's committees. Graduate students take courses, from within the department (in which

case a department staff member teaches the course), from outside the department, and even receive credit from courses taken at other institutions. Graduate students may serve as research assistants to staff members, or they may teach courses. As a result of further analysis, it became apparent that to build a graduate student record system in isolation from the rest of the department records would be both ineffective and inefficient.

The Project Leader prepared a proposal suggesting the development of an instructional department computer support system in which there would be four files:

1. A *Graduate Student Record File* which would contain the information on each graduate student registered in the department.
2. A *Staff Record File* which would contain information on staff members who have:
   a. taught courses in the department
   b. served on thesis committees in the department
3. A *Course File* which would contain information on the courses taught under the auspises of the department.
4. A *Mark File* which would contain a record of the marks obtained by all students (both graduate and undergraduate) taking courses from the department.

After defining the contents of each of these files, the Project Leader and the Department Chairman selected a group of department members who seemed especially interested in the project. These Key Contact Personnel included the

Department Chairman, the Department Secretary, the Administrative Assistant and three academic staff members. The Project Leader met with each of these individuals to ask for suggestions for improvement and additions to the information collected from the student records and interviews. The Key Contact Personnel were also asked to identify any possible extensions to the proposal which would lead to a more ideal system. The Key Contact Personnel suggested:

1. the addition of a section to allow for the maintenance of information from follow-up questionnaires sent to students who had graduated.
2. the inclusion of a date with many data elements in the Graduate Student Record File to allow for longitudinal studies.
3. the inclusion of a section on "reason for refusal to admit."

Each of these suggestions was incorporated into the IDCSS.

Once the proposal was modified to include the suggestions from the Key Contact Personnel, the Project Leader presented the proposed system to other administrators in the Faculty of Education and to staff involved in student records. These included the Assistant Dean in charge of student records, three department chairmen, two department secretaries and two department administrative officers. A few added capabilities were suggested by these people, however they appeared more interested in seeing if the IDCSS

would work as proposed.

The Project Leader then presented this proposal to a meeting of most of the academic staff in the Department of Educational Administration to ask the staff members to make suggestions for improvements to, or to identify any concerns with the system. Discussion arose concerning some additional information which might be included (medical information was one case), but the consensus was that no additions to the information beyond that presented in the proposal should be made. Some department members questioned the security of the system but their concerns appeared to be satisfied with the explanation that only three people, the Project Leader, the Department Secretary and the Department Chairman would have access to the system. A consensus was reached that the proposal was worthwhile and the project should proceed. Appendix H contains the definition of the structure of the prototype system as approved by the staff members and the Department Chairman.

The Department Chairman and the Project Leader then met to establish the scope of the system. The following constraints were established:

1. The prime function of the IDCSS would be to provide a graduate student record system.
2. Staff records would be limited to that information necessary to interact with the graduate student record file or to provide simple clerical assistance such as information used in preparation of the *Department of*

Educational Administration Student Information Brochure.

3. Access to the IDCSS would be limited to the Project Leader, the Department Secretary, and the Department Chairman.
4. Information and reports generated from the IDCSS would be made available to staff members only through the Department Chairman's office.
5. The Project would be funded by the department to the extent of necessary computing funds and clerical assistance.

4.3 Stage 4 - Design System

The prototype IDCSS accepted by the department members was subjected to further analysis in order to arrive at a design for the first implementation of the Department of Educational Administration Computer Support System (DEACSS). The following general rules were established for the design of this initial system.

1. No priviledged access to any particular information would be encoded into the actual definition of the file. This was deemed unnecessary since the only people with access to the file would be the Department Chairman, the Department Secretary, and the Project Leader. It was assumed that the security and privacy which was afforded the paper file system by the personal integrity of these three people would be transferred to the computerized system.

2. A minimal amount of coding of data (as defined in Section 3.3.3) would be performed during the initial implementation. It was felt that the flexibility allowed by maintaining data in the original form was preferable to the savings accrued from coding the data.
3. Since it was unknown what data might be missing for any student, the number of required data elements would be kept to a minimum.
4. Data would not be removed from a file unless they were incorrect.
5. All data would be placed in the file in reverse temporal order. In this way, the most recent value for each data element would be displayed as the first value for that data element.
6. Unless it was known for an certainty that a data element had a fixed length (e.g. Social Insurance Number), all data elements would be defined of variable length.

Appendix I gives a summary of the Data Element Dictionary for the prototype design of DEACSS.

4.4 Stage 5 - Implement System and Stage 6 - Formative Evaluation

The implementation and the formative evaluation of DEACSS were very closely interrelated. The file structure

was implemented as specified. The protocols[12] to enter a student's personal data were implemented and tested. As soon as the Department Secretary began to use these protocols on real student data a few problems with the design of the protocols became apparent. These problems were almost all of a minor nature to correct technically, but many were perceived by the Department Secretary to be major problems when it came to adding student data. As an example, the original specifications stated that the name, age and relationship of each student's dependent would be entered. This information would be used in helping to establish a student's need for financial assistance. It was found that the total number of dependents for each student was available, but not specific information on each dependent. The Department Secretary indicated frustration with having to enter "dummy" information for each dependent, when the only information she had was the total number of dependents. The file structure and the protocols were modified to enter only the number of dependents with an option to enter names and ages. Other problems of a similar nature were corrected as they were encountered.

Once the trial data were entered, a few sample reports were generated. These were mainly lists of students enrolled

----

[12] A SPIRES Protocol is a set of SPIRES commands which allow a programmer to provide a controlled environment in which a user who is naive of actual SPIRES commands can be allowed to perform many complex tasks based on the user's response to simple interactive questions and answers. DEACSS is completely controlled through protocols, hence the naive user need know no SPIRES commands.

in certain programs, with optional output of the students' address, program advisor, etc. From evaluating these reports, examining the data as it was actually stored in each file, and discussing the input Protocols with the Department Secretary, it was established that the system was working as originally intended, therefor the personal information for the remainder of the currently enrolled masters and doctoral students was entered.

Entry of the personal information on all the currently enrolled students was completed just previous to the 1979-80 registration period. The Project Leader and the Department Chairman felt that this would be a good time to test the system by generating certain reports for inclusion in the *Department of Educational Administration Student Information Brochure*. The Student/Advisor List (DEA/S/01) and the Advisor/Committee Member List (DEA/F/02) were included in the brochure. Examples of both these reports are included in Appendix E. At this time, the Department Secretary noted that there was a fair amount of staff information in this brochure which was retyped each year with a very few minor modifications. She suggested that this information should be entered into DEACSS so that the reports could be automatically generated each year with the most current staff information. Since this information was within the mandate of DEACSS, the *Staff File* was modified to include staff rank, office number, office telephone number, home address, home phone, and research interests. Two reports,

the Staff Address List (DEA/F/03) and the Staff Interest List (DEA/F/04) were generated. The first was distributed to all staff, while the second was placed in the *Department of Educational Administration Student Information Brochure*. Examples of these reports are also included in Appendix E.

At this time course marks were added to DEACSS. Course marks normally occur as groups six times a year. The Project Leader felt that it would be more efficient to keypunch the course and mark information, then add them as a batch job rather than to add each mark from the terminal. A transformation program was built to read keypunched cards and to build the appropriate SPIRES commands to add the mark records. Records were added to the *Mark File* and pointers were generated in the *Graduate Student Record File* and the *Course File* which pointed to the appropriate records in the *Mark File*. It soon became apparent that, while this method of maintaining student marks in a file separate from their personal information was efficient, it was far from effective. Because SPIRES stores pointer information in a form which is encoded, it was almost impossible for the Department Secretary to associate student mark information with the personal information.

Since one of the major priorities of the system was to make the system easy to use, the Project Leader decided to completely remove the *Mark File*, and to place mark information in the *Graduate Student Record File*. While this was less efficient, it certainly increased the effectiveness

of DEACSS as the marks for each course taken by each student now appeared in the *Graduate Student Record File*.

This change in design meant that all the data records had to be removed from SPIRES, the file structures redefined, and all the data records added back into the new system. As has been stated earlier, one of the major design criteria of SPIRES was that this kind of change must be possible with a minimum expenditure of both computer cost and human time. This change was accomplished in one night at a cost of approximately $150.00 in computing charges. Approximately ten person hours were spent respecifying and redesigning the system to implement this change.

There were many other small design and implementation changes, most of which had no effect on the data existing in the current data base. One other major design change occurred when a staff member (not one of the Key Contact Personnel) asked if it would be possible to generate a department instruction load report. This was within the scope of the project, so it was undertaken. The Department of Educational Administration offers a number of individual study and individual project courses. In order to keep the number of such course sections manageable, previous policy in the department was to assign many students taking an individual study course (e.g. ED ADM 591) to a single section. The teaching load for this section was then nominally assigned to a single staff member.

In order to generate a department instruction load report, each course had to be assigned to the actual staff member working with the student. In order to accommodate this change, the *Course File* and the *Graduate Student Record File* had to be modified to accept an extra character in the Course Section data element to indicate from which instructor each student was taking this course. One of the few data elements which was known "absolutely would not change" in size was the Course Section, hence it had been defined to be of fixed length! This modification was again performed in one night without affecting system performance, although the actual specification and design changes to allow for this modification took thirty two person hours. An example of the Department Instruction Load Report (DEA/F/05) is included in Appendix E.

By the middle of September 1979, the personal data on the active Ph. D. and M. Ed. students had been entered. As the year progressed, members of the department asked for different reports as the need became apparent. Each time a different report type was generated, a SPIRES format[13] was written to generate the report in a form which was professional in appearance. The SPIRES commands to select the correct students or staff members to be output in each report were included in a SPIRES protocol and were made part

----

[13] SPIRES Formats are the method used to generate reports from one or more SPIRES Subfiles. Formats are generally able to provide much more aesthetically pleasing output than normal default SPIRES output.

of DEACSS. Examples of all the reports which can be generated by DEACSS are given in Appendix E. The actual SPIRES formats which generate those reports and the SPIRES protocols which provide the interactive capabilities of DEACSS are documented in the *DEACSS Technical Manual* available from the Department of Educational Administration.

In June 1980, the Project Leader and the Department Chairman decided that a summative evaluation of DEACSS needed to be undertaken. Both felt that it was too early for a summative evaluation, but a number of concerns indicated that a summative evaluation should be undertaken:

1. a new winter session would begin in two to three months, this was the time when the majority of student counselling was undertaken in the department. Delaying meant that at least one more year would go by before DEACSS could be tested under such circumstances.
2. a number of educational agencies both within and without the Faculty of Education had been asking for a report on the status and availability of DEACSS. It was felt that some closure should be attempted for these other agencies.

In order to facilitate summative evaluation, the *DEACSS User's Manual* (Appendix A) was completed which gives the necessary instructions, both at the simple user level and at the most advanced user level, to use and manage DEACSS.

The structure of this "final" version of DEACSS is given in Appendix C, while the summarized Data Element

Dictionary is given in Appendix D. When the summative evaluation was begun, the reports shown in Table 4.1 were available. Options one to six, those options which were student oriented, were each available for the groups of students shown in Table 4.2. A procedure which allowed output from DEACSS to be used in TEXTFORM, the text formatting language available on MTS, was also defined. Samples of the TEXTFORM commands to generate both a form letter and a questionnaire are included in Appendix F. Example output from these letters are also included in that appendix.

4.5 Stage 7 - Summative Evaluation

In August 1980, the Department Chairman, Department Secretary, Administrative Assistant and the Programmer Analyst in the Department of Educational Administration were provided with a copy of the *DEACSS User's Manual* (Appendix A). Approximately one week later, DEACSS was demonstrated to these people. After demonstration, each person was asked to comment on the system.

In September 1980 each member of the academic staff was provided with two copies of the complete student record as held in DEACSS for each student that member advised plus copies of the student related reports itemized in Table 4.1. They were asked to use these data when counseling students on their graduate programs. Each student was to be provided

Table 4.1 Report Options for DEACSS

Option	Report Number	Contents of Report
1	DEA/S/01	Copies of the complete student file for a group of students .
2	DEA/S/02	A list of student addresses.
3	DEA/S/03	A list of students with advisors.
4	DEA/S/04	A list of student addresses and advisors.
5	DEA/S/05	A list of students with no advisor.
6	DEA/S/06	A list of all students requesting assistance.
51	DEA/F/01	A staff list with rank, office and phone.
52	DEA/F/02	A list of staff with their advisees, and students on whose committees they serve.
53	DEA/F/03	A staff list with home address and phone.
54	DEA/F/04	A staff list with interest, office and phone.
55	DEA/F/05	A Department Instruction Load Report.

Table 4.2 Optional Groupings for DEACSS Student Reports

Option	Groups of Students for Reports
1	Full Time Ph.D.
2	Part Time Ph.D.
3	All Ph.D.
4	Full Time Masters
5	Part Time Masters
6	All Masters
7	Thesis Masters
8	Non Thesis Masters
9	Administrative Development Program
10	Teaching Skills Improvement Program (Full Time)
11	Full Time Masters (no ADP or TSIP)
12	Part Time Masters (no ADP or TSIP)
13	All Masters (no ADP or TSIP)
14	All Active Students

with a copy of his or her own file, and they were asked to report any errors or omissions to the Department Secretary.

After student registration was completed, six staff members were interviewed. They were asked to comment on DEACSS with respect to four questions:

1. How useful were the Student Files provided during registration?
2. How can the information in the Student Files be improved?
3. Should DEACSS be continued or should it be discontinued?
4. If DEACSS should be continued, what areas would you like to see added to the system?

The responses to these questions are given in Chapter six.

The system was also demonstrated to all six Department Chairpersons, four Department Secretaries, and three Administrative Professional Officers or Administrative Assistants, and one Assistant Department Chairman in the Faculty of Education. Administrative personnel from outside the Faculty of Education requested and received demonstrations of DEACSS. These included:

1. the Administrative Assistant to the Dean of Graduate Studies and that faculty's Programmer Analyst,
2. a representative of the University of Victoria, and
3. three members of the Registrar's office of Concordia College.

Comments and evaluations received during these demonstrations will be given in Chapter six.

4.6 A Path Diagram of DEACSS

Figure 4.1 shows a path diagram of the development of DEACSS from the Perception of Need for a Computer Support System stage Summative Evaluation stage. Standard notation for path diagrams is followed with paths between nodes indicating an activity and the nodes indicating the start and/or completion of an activity or activities. The span of each of the major stages has been indicated on Figure 4.1. The dates for important nodes have been given to give an idea of the time frame for this development.

The following sections discuss each of the activities within each of the stages in the development of DEACSS.

4.6.1 Perception of Need for a Computer Support System

Time span: January 1, 1978 to February 1, 1979.

Path	Activity
1 - 2	**Perception of Need** The Department Chairman, the Project Leader and other members of the Department discussed the general feasibility of and the possible facilities to be provided by an IDCSS for the Department of Educational Administration.

Figure 4.1 Path Diagram for the Development of DEACSS

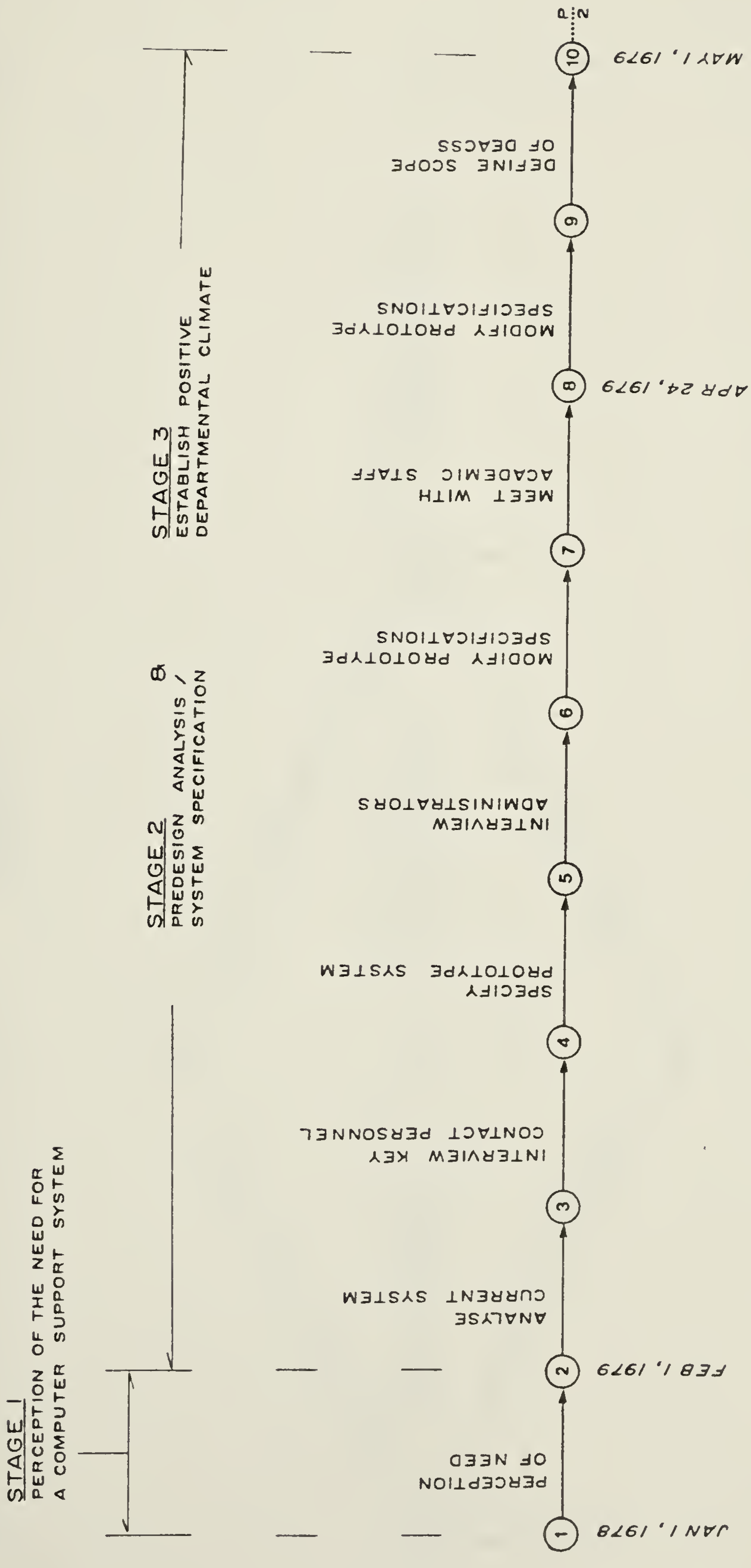


Figure 4.1 Path Diagram for the Development of DEACSS
(continued)

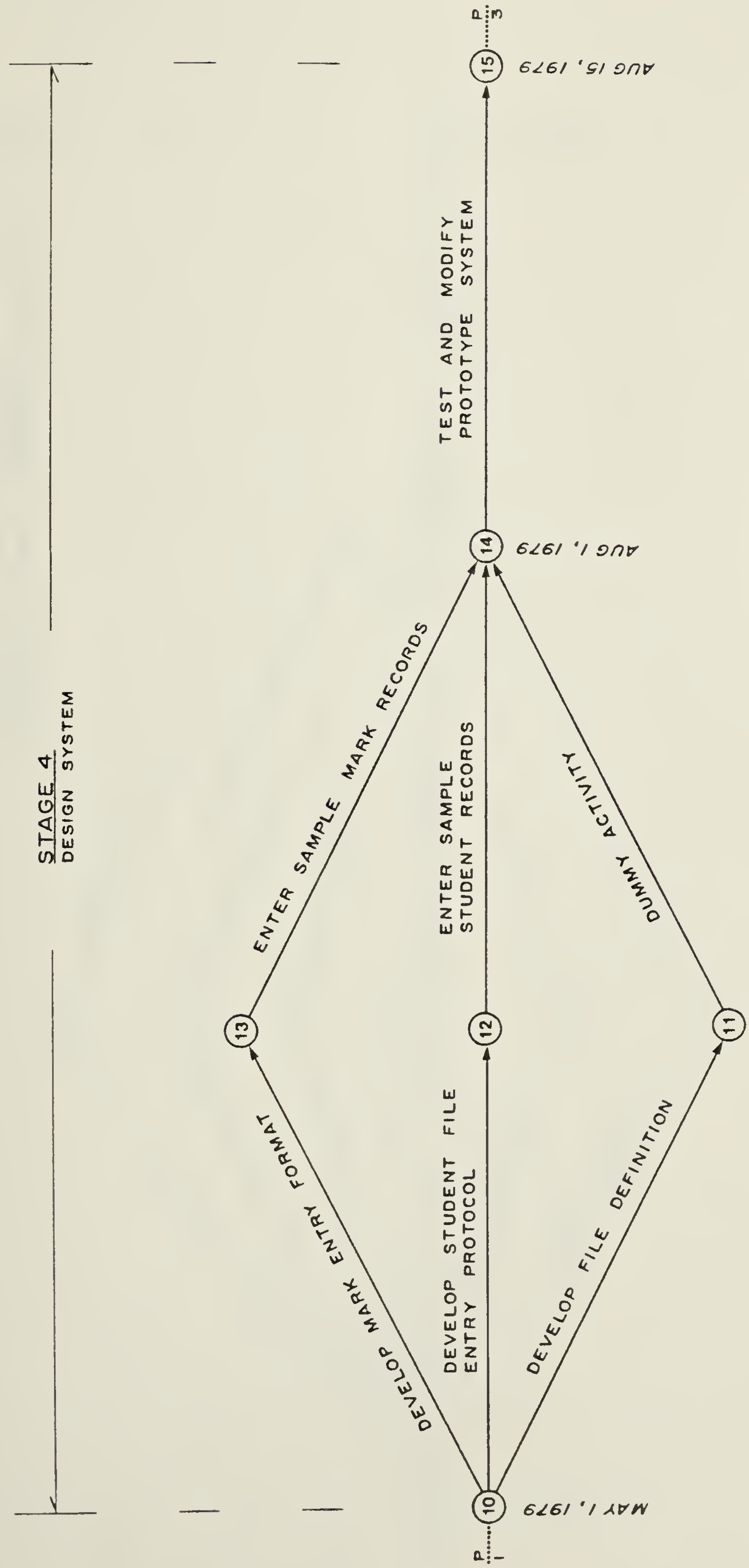


Figure 4.1 Path Diagram for the Development of DEACSS (continued)

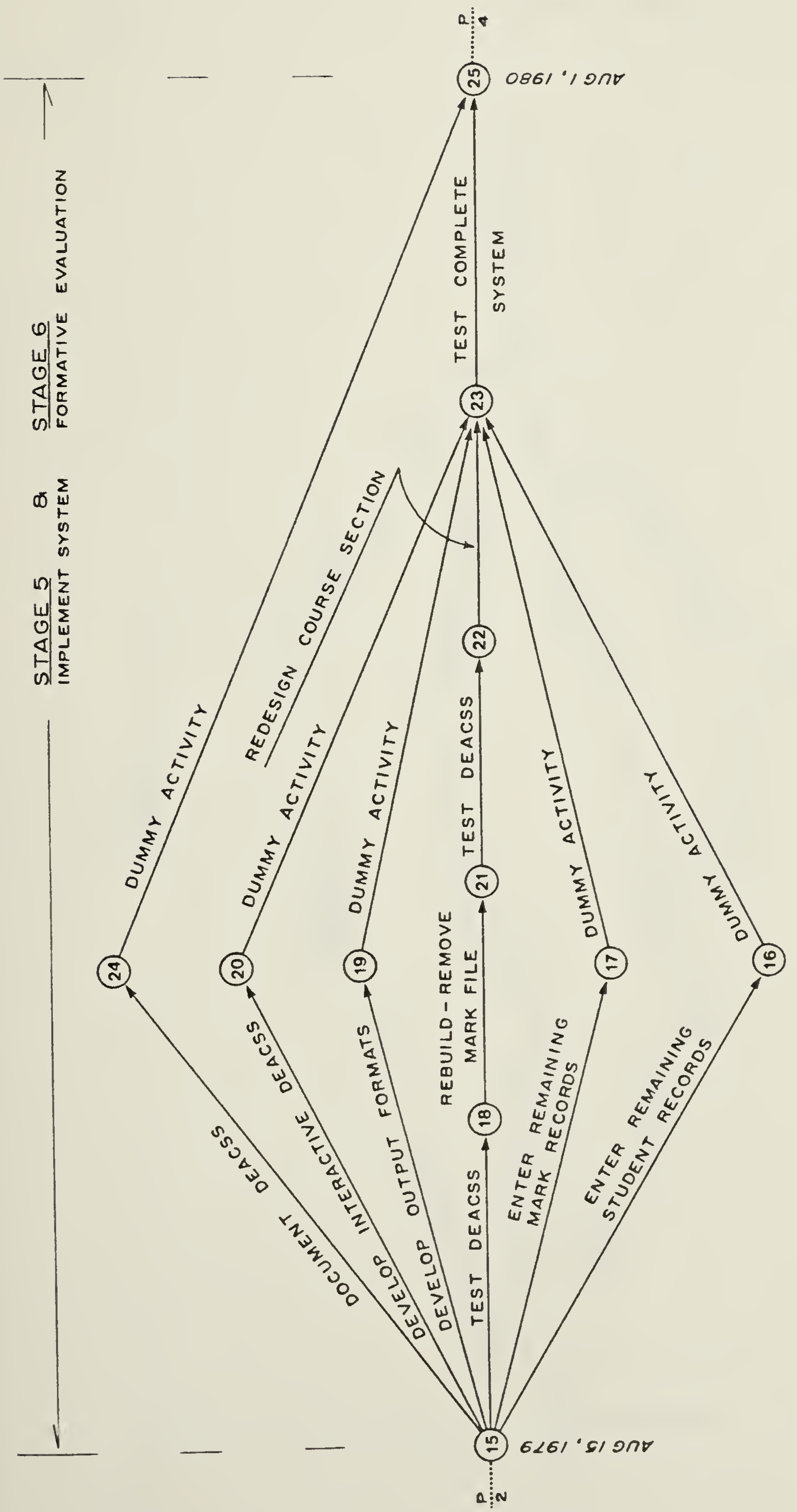
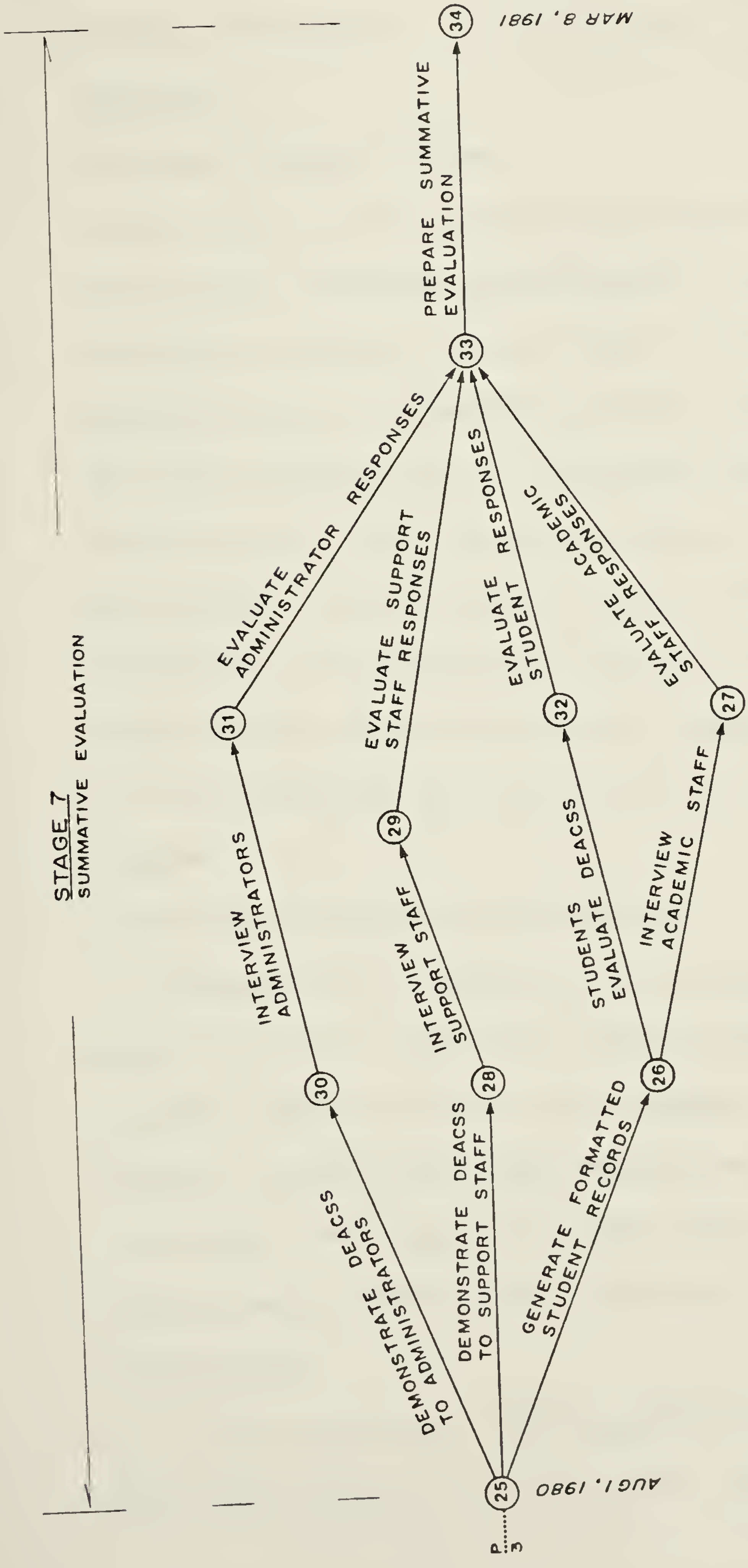


Figure 4.1 Path Diagram for the Development of DEACSS (continued)



4.6.2 Predesign Analysis/System Design and Establish Positive Department Climate

Time span: February 1, 1979 to May 1, 1979.

Path	Activity
2 - 3	**Analyse Current System**
	An analysis of the current methods of storing and retrieving information was made. The kinds of information kept, uses of that information and possible other information which had been required but was found to be not available was also studied. This analysis was based upon the information maintained in the current paper system in the Department of Educational Administration, the requirements specified by the Department Chairman, and the elements maintained by other student record systems.
3 - 4	**Interview Key Contact Personnel**
	The Department Secretary and the department Administrative Assistant, both closely involved in student records, were interviewed regarding the format in which the data should be entered and retrieved. Through this, some data elements were identified as having been omitted from those already identified.
	Three academic staff members were interviewed about what they felt a computer support system

should do for administrative and academic staff.

4 - 5 **Specify Prototype System**

A prototype structure for a computer support system was designed based upon the information obtained in steps 1-2, 2-3 and 3-4.

5 - 6 **Interview Administrators**

Members of the administration and planning groups within the Faculty of Education were interviewed. The interview was structured to examine each area of the prototype system, asking for additions and changes. As well as specifying what information should be stored, they were be asked to specify the form and format in which it should be retrieved.

6 - 7 **Modify Prototype Specifications**

The specifications for the prototype system were modified to include the suggestions from the administrators and planners.

7 - 8 **Meet with Academic Staff**

A meeting with most of the academic staff in the Department of Educational Administration was structured to examine all areas of the proposed system, asking for recommendations as to where the system should be expanded, modified or restricted.

8 - 9 **Modify Prototype Specifications**

The specifications for the prototype system were modified to include the suggestions from the academic staff.

9 -10 **Define Scope of DEACSS**

The Project Leader and the Department Chairman met to discuss department resources and those features of the prototype system which should be implemented. At this time it was decided to attempt to implement the complete prototype system and to see how the system evolved.

4.6.3 Design System

Time span: May 1, 1979 to August 15, 1979.

Path Activity

10-11 **Develop File Definition**

The file definition for the prototype of DEACSS was defined to meet the specifications. This definition was then programmed in the SPIRES File Definition language, implemented and tested.

11-14 **Dummy Activity**

10-12 **Develop Student File Entry Protocol**

A protocol which facilitated entry of student records by responding to prompts was designed and implemented.

12-14 **Enter Sample Student Records**

The records from a small sample of students were entered as a test.

10-13 **Develop Mark Entry Format**

A keypunching procedure was established to transfer course registration and mark information from

official mark reporting forms. A FORTRAN program was written which reformatted this data so it could be easily entered into the SPIRES file.

13-14 **Enter Sample Mark Records**

The course registration and mark records for one academic year were keypunched and entered.

14-15 **Test and Modify Prototype System**

The prototype system was tested to see if it worked as specified. Modifications were made until the prototype system met design specifications. At this point the system was given the name DEACSS.

4.6.4 Implement System and Formative Evaluation

Time span: August 15, 1979 to August 1, 1980.

Path Activity

15-18 **Test DEACSS**

DEACSS was tested by using it to answer questions which occurred in the day to day operation of the department. While the data was not complete, the ability of the system to meet the needs of the department were evaluated.

18-21 **Rebuild DEACSS - Remove Mark File**

The *MARK* file was removed from the system. Marks were entered directly into the *Graduate Student Record File* for each student. This made it easier for the user to see all information on one student in one place.

21-22 **Test DEACSS**

DEACSS was constantly monitored to see how well it achieved the original design objectives.

22-23 **Redesign Course Section**

The Course Section was redesigned to include a sub section in order to facilitate design of the Staff Instruction Load Report.

15-16 **Enter Remaining Student Records**

The remainder of the student records for the currently enrolled students were entered into the system.

16-23 **Dummy Activity**

15-17 **Enter Remaining Mark Records**

Course Marks for the years 1977 to present were keypunched and entered into the system.

17-23 **Dummy Activity**

15-19 **Develop Output Formats**

The output formats for the requested reports were designed and implemented.

19-23 **Dummy Activity**

15-20 **Develop Interactive DEACSS**

A series of protocols were developed which led the user through most common facilities of DEACSS via responses to prompts.

20-23 **Dummy Activity**

23-25 **Test Complete System**

The complete system was tested using both native

SPIRES and the SPIRES protocols which were developed as Interactive DEACSS. A few modifications were made to the format of reports and to the way data was entered or searched.

15-24 **Document DEACSS**

A User's Manual for DEACSS was prepared.

24-25 **Dummy Activity**

4.6.5 Summative Evaluation

Time span: August 1, 1980 to March 8, 1981.

<u>Path</u> <u>Activity</u>

25-26 **Generate Formatted Student Records**

A formatted student record was generated for each currently registered student. Copies of these were given to the academic staff to use in counseling students during the 1980-81 registration period.

26-27 **Interview Academic Staff**

Academic staff members were interviewed asking for an evaluation of the system.

27-35 **Evaluate Academic Staff Responses**

The responses from the academic staff members were collated and analysed. A report of academic staff members' perceptions of DEACSS is included in Chapter six.

25-28 **Demonstrate DEACSS to Support Staff**

DEACSS was demonstrated to support staff in the department who would be most involved with the use

of the system. This included the department Administrative Assistant, the Department Secretary (who had just assumed this position), and the department Computer Programmer.

28-29 **Interview Support Staff**

The support staff in the department were interviewed and asked to evaluate DEACSS.

29-35 **Evaluate Support Staff Responses**

The responses from the support staff members were collated and analysed. A report of support staff members perceptions of DEACSS is included in Chapter six.

25-30 **Demonstrate DEACSS to Administrators**

The system was demonstrated to administrators and planners both within and outside the faculty. Representatives of two other post secondary institutions also received demonstrations of DEACSS.

30-31 **Interview Administrators**

The administrators and planners to whom DEACSS was demonstrated were asked to evaluate the system.

31-33 **Evaluate Administrator Responses**

The responses from the administrators and planning staff were collated and analysed. A report of the administrator and planning staff perceptions of DEACSS is included in Chapter six.

26-32 **Students Evaluate DEACSS**

A questionnaire and a copy of his or her DEACSS

student record was given to each student by his or her advisor. Students were asked to check the veracity of the DEACSS student record and were asked about their background concerning computers and their opinions about computerized student records.

32-33 **Evaluate Student Responses**

The responses from the students were collated and analysed. A report of student perceptions of DEACSS is included in Chapter six.

33-34 **Prepare Summative Evaluation**

A final report on the project in the form of a thesis was written.

5. A COST ACCOUNTING FOR DEACSS

5.1 Method of Reporting Costs

A cost accounting was maintained during all stages of DEACSS. These figures are presented so that an organization wishing to implement a similar system can estimate its costs. The hourly costs of different types of personnel varies widely from institution to institution and over time. Therefore, the contribution of individuals working on the project will be noted in terms of "person hours" for each class of individual. Equipment costs will be given in terms of dollars. Computer costs will be given in terms of dollars.

An explanation of the applicability and categorizations of each of these measures follows.

5.1.1 Person Hour Costs

The time contribution of individuals working on, or consulting to, the project was tabulated in terms of person hours.

Five categories of people worked on, or were consulted in conjunction with the project:

1. *Project Leader/System Designer*

   The author acted as the Project Leader and the System Designer. This job entailed the predesign analysis, the synthesis, the design evaluation and the management of the project.

2. *Computer Analyst*

The author also acted as the major computer analyst. When other computer analysts contributed to the project, their time was incorporated under this heading.

Division of time between the job of System Designer and that of Computer Analyst was sometimes artificial, but the times given here are a fair representation of the split in time. It must be stressed that the sharing of the two jobs by the same individual certainly reduced time that would be spent by the System Designer clarifying ideas with the Computer Analyst. The time saved was probably equivalent to that lost due to the fact that the author was learning SPIRES as the project progressed, hence was not an efficient programmer.

3. *Department Secretary/Data Entry*

The Department Secretary undertook the major task of finding and entering appropriate information into DEACSS. This job could have been split between the Department Secretary doing the "thinking" tasks and a clerk typist doing the data entry, but the savings would probably not have been great.

4. *Keypunch Operator*

Class enrollments for each course in Educational Administration were keypunched immediately after the beginning of each term. Once the term was completed, student marks were keypunched. Since these data occurred in two large spurts each term, and since the data could

be punched by a keypunch operator with no knowledge of DEACSS, using this method of entering these data saved much time for the Department Secretary.

5. *System Specification Consultants*

   Other people were consulted during the design, implementation, and evaluation of the system. The kind of individual consulted will be identified in the following sections.

5.1.2 Equipment Costs

Two major pieces of equipment were obtained by the department for use on the project. DEACSS was designed to be run as an interactive program. This necessitated that the department provide a computer terminal with very powerful "visual editing" capabilities. The department already had leased such a terminal for other uses in the department, and this terminal was borrowed as necessary. This turned out to be a very unsatisfactory arrangement: at times it was necessary to use the terminal for long periods which made it unavailable for the original purposes, while at other times the terminal was unavailable for use on DEACSS when needed.

Any unit intending to implement a system similar to DEACSS should plan to purchase or lease a terminal to be dedicated to this application. Currently, the recommended terminal on the University of Alberta campus is the Anderson Jacobson 510 terminal, modified by the Department of Computing Services to have visual editing capabilities.

This terminal can be purchased for $3100.00 with monthly maintenance and communications charges of $31.25. This same terminal can be leased for a cost of $112.00 per month inclusive of maintenance and communications charges.

A second terminal, an Anderson Jacobson 832 "letter quality" printing terminal, was used for direct output of letters, envelopes and forms; and was borrowed as necessary from another department. While such a terminal is not imperative during the development stages of the project, as the system becomes operative it must be expected that such a terminal will become more necessary.

A terminal with similar features can be purchased for approximately $5000.00 with monthly maintenance and communications charges of $31.25. This same terminal can be leased for a cost of $105.00 per month inclusive of maintenance and communications charges.

It should be noted that the prices given above are quoted in Canadian dollars and were prices available to the University of Alberta in 1980. They should be used as representative figures only.

5.1.3 Computer Costs

This system was designed and implemented on an Amdahl 470 V/7 computer operating under the Michigan Terminal System (MTS). A copy of the charging system and costs for using MTS during 1980-81 is shown in Appendix G.

5.2 Cost Breakdown

The following section reports costs involved in the development of DEACSS. During the project, information was itemized according to the paths shown on Figure 4.1 and described in Section 4.6. It seems reasonable that future implementations will follow a similar course, hence costs will be reported for each of these paths. Costs will be itemized by job undertaken, individuals involved and time or computer costs involved. Except where noted, all person hours accounted were provided by members of the Department of Educational Administration or, in the case of technical assistance, paid for by the Department of Educational Administration. Individuals outside the department donated time to the project in assisting in the specification of, and the evaluation of DEACSS. Time donated by people outside the Department of Educational Administration is noted in the following section with an asterisk (*).

5.2.1 Stage 1 - Perception of Need for a Computer Support System

Time span: January 1, 1978 to February 1, 1979.

Path	Activity
1 - 2	**Perception of Need**
	Project Leader/System Designer...... 10 Person Hours
	Department Secretary/Data Entry...... 1 Person Hours
	System Specification Consultants

Department Chairman.............. 5 Person Hours

Academic Staff................... 2 Person Hours

5.2.2 Stage 2 - Predesign Analysis/System Specification and Stage 3 - Establish Positive Department Climate

Time span: February 1, 1979 to May 1, 1979.

Path	Activity
2 - 3	**Analyse Current System**
	Project Leader/System Designer...... 33 Person Hours
	Department Secretary/Data Entry...... 2 Person Hours
	System Specification Consultants
	Department Chairman.............. 4 Person Hours
3 - 4	**Interview Key Contact Personnel**
	Project Leader/System Designer....... 8 Person Hours
	Department Secretary/Data Entry...... 4 Person Hours
	System Specification Consultants
	Administrative Officer........... 1 Person Hours
	Academic Staff................... 3 Person Hours
4 - 5	**Specify Prototype System**
	Project Leader/System Designer...... 56 Person Hours
5 - 6	**Interview Administrators**
	Project Leader/System Designer..... 9.5 Person Hours
	System Specification Consultants
	Department Chairpersons*......... 7 Person Hours
	Administrative Officers*......... 3 Person Hours
	Planning Staff*................ 2.5 Person Hours
	Department Secretaries*.......... 3 Person Hours

6 - 7 **Modify Prototype Specifications**

Project Leader/System Designer....... 3 Person Hours

7 - 8 **Meet with Academic Staff**

Project Leader/System Designer....... 2 Person Hours

System Specification Consultants

Academic Staff.................. 18 Person Hours

8 - 9 **Modify Prototype Specifications**

Project Leader/System Designer....... 2 Person Hours

9 -10 **Define Scope of DEACSS**

Project Leader/System Designer....... 3 Person Hours

System Specification Consultants

Department Chairman.............. 3 Person Hours

Summary of Costs for Predesign Analysis / System Specification and Establish Positive Department Climate Stages

Project Leader/System Designer... 116.5 Person Hours

Department Secretary/Data Entry...... 6 Person Hours

System Specification Consultants

Department Chairman..............9 Person Hours

Administrative Assistant..........1 Person Hours

Academic Staff.....................21 Person Hours

Department Chairpersons*..........7 Person Hours

Administrative Officers*......... 3 Person Hours

Planning Staff*................ 2.5 Person Hours

Department Secretaries*......... 3 Person Hours

5.2.3 Stage 4 - Design System

Time span: May 1, 1979 to August 15, 1979.

Path	Activity
10-11	**Develop File Definition**
	Project Leader/System Designer....... 5 Person Hours
	Computer Analyst.................... 82 Person Hours
	Computer Costs..............................$300.00
11-14	**Dummy Activity**
10-12	**Develop Student File Entry Protocol**
	Project Leader/System Designer....... 4 Person Hours
	Computer Analyst.................. 16.5 Person Hours
	Department Secretary/Data Entry...... 1 Person Hours
	Computer Costs..............................$100.00
12-14	**Enter Sample Student Records**
	Project Leader/System Designer....... 2 Person Hours
	Department Secretary/Data Entry..... 10 Person Hours
10-13	**Develop Mark Entry Format**
	Project Leader/System Designer....... 2 Person Hours
	Computer Analyst.................... 20 Person Hours
	Keypunch Operator.....................6 Person Hours
	Computer Costs...............................$10.00
13-14	**Enter Sample Mark Records**
	Project Leader/System Designer....... 2 Person Hours
	Computer Costs...............................$20.00
14-15	**Test and Modify Prototype System**
	Project Leader/System Designer...... 10 Person Hours
	Computer Analyst.................... 71 Person Hours

Department Secretary/Data Entry...... 5 Person Hours

Computer Costs..............................$100.00

Summary of Costs for Design System Stage

Project Leader/System Designer...... 25 Person Hours

Computer Analyst................ 189.5 Person Hours

Department Secretary/Data Entry..... 16 Person Hours

Keypunch Operator.....................6 Person Hours

Computer Costs..............................$530.00

5.2.4 Stage 5 - Implement System and Stage 6 - Formative Evaluation

Time span: August 15, 1979 to August 1, 1980.

Path Activity

15-18 **Test DEACSS**

The operation of DEACSS was constantly being monitored by the Department Chairman, the Department Secretary and the Project Leader during all procedures. No special costs can be attributed to this step.

18-21 **Rebuild DEACSS - Remove Mark File**

Project Leader/System Designer....... 1 Person Hours

Computer Analyst..................... 8 Person Hours

Department Secretary/Data Entry...... 1 Person Hours

Computer Costs..............................$150.00

21-22 **Test DEACSS**

The operation of DEACSS was constantly being monitored by the Department Chairman, the Department

Secretary and the Project Leader during all procedures. No special costs can be attributed to this step.

22-23 **Redesign Course Section**

Project Leader/System Designer....... 5 Person Hours

Computer Analyst.................... 25 Person Hours

Department Secretary/Data Entry...... 1 Person Hours

System Specification Consultants

Academic Staff Member............ 2 Person Hours

Computer Costs.............................$250.00

15-16 **Enter Remaining Student Records**

Project Leader/System Designer...... 15 Person Hours

Department Secretary/Data Entry.. 301.5 Person Hours

Computer Costs............................$1701.00

16-23 **Dummy Activity**

15-17 **Enter Remaining Mark Records**

Project Leader/System Designer....... 5 Person Hours

Computer Analyst.................... 30 Person Hours

Keypunch Operator...................48 Person Hours

Computer Costs.............................$300.00

17-23 **Dummy Activity**

15-19 **Develop Output Formats**

Project Leader/System Designer...... 19 Person Hours

Computer Analyst.................... 97 Person Hours

Computer Costs.............................$400.00

19-23 **Dummy Activity**

15-20 **Develop Interactive DEACSS**

Project Leader/System Designer...... 23 Person Hours

Computer Analyst.................. 135 Person Hours

Computer Costs.............................$800.00

20-23 **Dummy Activity**

23-25 **Test Complete System**

Project Leader/System Designer....... 8 Person Hours

Computer Costs..............................$45.00

15-24 **Document DEACSS**

Project Leader/System Designer..... 185 Person Hours

Computer Costs.............................$500.00

24-25 **Dummy Activity**

Summary of Costs for Implement System and Formative Evaluation Stages

Project Leader/System Designer.... 261 Person Hours•

Computer Analyst.................. 295 Person Hours

Department Secretary/Data Entry. 302.5 Person Hours•

Keypunch Operator...................48 Person Hours

System Specification Consultants

Department Chairman............ 0 Person Hours•

Academic Staff Member........... 2 Person Hours

Computer Costs............................$4146.00

• Estimates are low since time monitoring DEACSS during normal operation could not be estimated.

5.2.5 Stage 7 - Summative Evaluation

Time span: August 1, 1980 to March 8, 1981.

Path	Activity
25-26	**Generate Formatted Student Records**
	Project Leader/System Designer....... 3 Person Hours
	Computer Costs..............................$40.00
26-27	**Interview Academic Staff**
	Project Leader/System Designer..... 5.5 Person Hours
	System Specification Consultants
	Academic Staff Member.......... 5.5 Person Hours
27-35	**Evaluate Academic Staff Responses**
	Project Leader/System Designer....... 3 Person Hours
25-28	**Demonstrate DEACSS to Support Staff**
	Project Leader/System Designer..... 1.5 Person Hours
	Computer Analyst..................... 2 Person Hours
	Department Secretary/Data Entry.... 1.5 Person Hours
	System Specification Consultants
	Administrative Assistant....... 1.5 Person Hours
	Computer Costs..............................$4.00
28-29	**Interview Support Staff**
	Project Leader/System Designer....... 1 Person Hours
	Computer Analyst..................... 1 Person Hours
	Department Secretary/Data Entry...... 1 Person Hours
	System Specification Consultants
	Administrative Assistant......... 1 Person Hours
29-35	**Evaluate Support Staff Responses**
	Project Leader/System Designer....... 1 Person Hours

25-30 **Demonstrate DEACSS to Administrators**

Project Leader/System Designer...... 12 Person Hours

System Specification Consultants

Department Chairman.............. 2 Person Hours

Department Chairpersons*......... 4 Person Hours

Planning Staff*.................. 1 Person Hours

Administrative Officers*......... 2 Person Hours

Department Secretaries*.......... 2 Person Hours

Non Fac. of Ed. Administrators*..14 Person Hours

30-31 **Interview Administrators**

Project Leader/System Designer....... 6 Person Hours

System Specification Consultants

D.E.A. Department Chairman....... 1 Person Hours

Department Chairpersons*....... 3.5 Person Hours

Planning Staff*.................. 1 Person Hours

Administrative Officers*......... 1 Person Hours

Department Secretaries*.......... 1 Person Hours

Non Fac. of Ed. Administrators*...6 Person Hours

31-33 **Evaluate Administrator Responses**

Project Leader/System Designer....... 2 Person Hours

26-32 **Students Evaluate DEACSS**

Project Leader/System Designer....... 4 Person Hours

System Specification Consultants

Students........................ --- Person Hours

32-33 **Evaluate Student Responses**

Project Leader/System Designer....... 8 Person Hours

Computer Analyst..................... 5 Person Hours

Department Secretary/Data Entry..... 20 Person Hours

Keypunch Operator....................20 Person Hours

Computer Costs.............................$20.00

33-34 **Prepare Summative Evaluation**

Project Leader/System Designer...... 80 Person Hours

Summary of Costs for Summative Evaluation Stage

Project Leader/System Designer..... 127 Person Hours

Computer Analyst..................... 8 Person Hours

Department Secretary/Data Entry....22.5 Person Hours

Keypunch Operator....................20 Person Hours

System Specification Consultants

Department Chairman............. 3 Person Hours

Academic Staff Members......... 5.5 Person Hours

Administrative Assistant....... 2.5 Person Hours

Department Chairpersons*....... 7.5 Person Hours

Planning Staff*................. 2 Person Hours

Administrative Officers*......... 3 Person Hours

Department Secretaries*.......... 3 Person Hours

Non Fac. of Ed. Administrators*..22 Person Hours

Computer Costs.............................$64.00

5.3 Summary of Costs for Development of DEACSS

A summary of all costs for DEACSS is provided in Table 5.1. In this table, all costs for humans are provided in person hours, while computing costs are given in MTS computer dollars.

Table 5.1 Summary of Costs for DEACSS

	STAGE 1	STAGES 2&3	STAGE 4	STAGES 5&6	STAGE 7	TOTAL
Project Director/ System Designer	10	116.5	25	261•	127	539.5•
Computer Analyst			189.5	295	8	492.5
Dept. Secretary /Data Entry	1	6	16	302.5•	22.5	348•
Keypunch Operator			6	48	20	74
Dept. Chairman	5	9		0•	3	17•
Admin. Assistant		1			2.5	4.5
Academic Staff	2	21		2	5.5	30.5
Dept.* Chairs.		7			7.5	14.5
Planning* Staff		2.5			2	2
Admin.* Officers		3			3	6
Dept.* Secretary		3			3	6
Non-Educ. Faculty*					22	22
Computer Charges			$530	$4146	$64	$4740

* Time volunteered by personnel from outside the Department of Educational Administration.

• Estimates are low since time monitoring DEACSS during normal operation could not be estimated.

6. EVALUATION OF DEACSS AND THE IDCSS STRATEGY

An evaluation of the strategy for the implementation of an IDCSS is an almost impossible task. One would have to implement the strategy many times in many different environments before one could come to the conclusion that the strategy would usually work. In this study the strategy was used once, to implement DEACSS. In the Summative Evaluation Stage, the staff members and administrators who were asked to assist in the evaluation were evaluating DEACSS, not the general strategy.

After evaluating this single implementation, inferences as to the usefulness of the strategy in general will be made. It is acknowledged that this case study approach has weaknesses and it is hoped that the reader does not infer that because the strategy worked successfully once, it will always work.

6.1 Evaluation of DEACSS

In September of 1980, it was decided by the Project Leader and the Department Chairman that DEACSS had reached a point where a full scale trial should be undertaken. It was also decided that due to time considerations a summative evaluation should be undertaken. Neither the Project Leader nor the Department Chairman felt comfortable that DEACSS had reached its full potential, but there were extenuating circumstances which forced an evaluation at this time:

1. it was the beginning of a new university session, therefore it was the best time to give the student record files to the staff members to be used in program counselling.
2. other departments and institutions had indicated interest in obtaining a copy of DEACSS to be modified for their own use hence some closure on the project needed to be made.

The evaluation of DEACSS was undertaken by eliciting information from five different groups of people:

1. a sample of academic staff in the department.
2. graduate students in the department.
3. a sample of the support staff in the department.
4. department chairpersons, and administrative support staff from agencies outside the Department of Educational Administration.
5. The Chairman of the Department of Educational Administration.

6.1.1 Evaluation of DEACSS by Department of Educational Administration Academic Staff

The academic staff members had been exposed to DEACSS, or more specifically to reports produced by DEACSS, in two major ways:

1. The *Department of Educational Administration Student Information Brochure* contained a number of lists of information deemed useful to both staff and students. A great deal of this information was produced directly

from DEACSS. This handbook was given to all staff and full time students in the department at the beginning of the 1980-1981 academic session.

2. Each staff member was provided with two copies of the student record for each student that member was advising. They then gave one copy of the student record plus an explanatory letter and a questionnaire[14] for each student. The questionnaire concerned the familiarity of the students with computers and their attitudes towards the computerization of their student record. The students were asked to complete the questionnaire and to check the copy of the student file for both completeness and veracity. Any corrections to the file plus the questionnaire were to be returned to the department office. The second copy of the student file was to be retained by the staff member to be used in advising the student on program choices.

This was the first time that staff were not given the complete department file folder on each student. In a personalized covering letter, each staff member was informed that if there was any question as to the completeness or the veracity of the information in the supplied student file, or if more information such as letters of reference was necessary, the official department student file folders were available in the

----

[14] Example copies of the personalized explanatory letters to both staff and students and the questionnaire are given in Appendix J.

general office. In the letters to both staff and students it was explicitly stated that the student files included only courses taken after the beginning of 1977, and that marks from Summer Session 1980 were not available. Many of the staff perceived this limitation as a very serious problem.

The usefulness of the DEACSS file in student counselling was raised in the discussion at the Department of Educational Administration Staff meeting of September 5, 1980. Under the heading of "*Computerized Student File*" the minutes of this meeting stated:

> "Interaction indicated that there are still major deficiencies in these files. Amendments and additional information will hopefully improve the files' value to advisors."

The majority of staff complaints concerned the incompleteness of the courses and marks in the students' files. After this meeting the Project Leader met with the Department Chairman, and five members of the staff who had expressed diverse views on the usefulness of DEACSS in student counselling were selected to be interviewed individually.

The in depth interviews with each of these staff members were intentionally unstructured. This gave the Project Leader the freedom to allow the interview to address many issues beyond those affecting the usefulness of the system in student counselling.

While the interviews were unstructured, the following

four major questions were posed in each interview.

1. When counselling students, how useful were the student records produced by DEACSS?
2. How can these records be improved to provide better information for student counselling?
3. Should DEACSS be continued in its current form, discontinued completely, or continued in some modified form?
4. If DEACSS should be modified, what should be added or deleted?

Although the five staff members had indicated diverse views at the staff meeting, a significant amount of disagreement in response to each of these questions did not materialize. Instead, staff members voiced similar opinions, but their opinions varied as to the importance of each concern raised.

6.1.1.1 When counselling students, how useful were the student records produced by DEACSS?

All five staff members interviewed identified the problem that the courses and marks in the DEACSS record were incomplete or inaccurate. This was seen as a major problem by four of the five members. The fifth stated that he did not perceive this as a major problem since he could ask the student for verification of the printed information. One of the staff members stated that because of this problem, he got the "real" file (i.e. the department student file

folder) for each student before the interview in order to be able to check information.

Although one of the design criteria for DEACSS had been that everything reasonably possible would be copied from the real file to the computer file, two staff members reported that they liked the real file for student consulting since it was "more complete". The real file, for example, included letters of reference which had not been included in the computer file. One of these two members said that he felt more comfortable using the "real" file, since the information in it was "prepared by the student." He liked to see the student's handwriting and indicated that he felt the information in the computer file was "cold" as it had been prepared by someone else.

6.1.1.2 How can the student file be improved to provide better information for student counselling?

Once again, every member interviewed identified the problem with the completeness and veracity of the students' marks. All members suggested that one of the most important considerations in the file should be that the courses should be complete and the marks correct. When it was pointed out that this is just as big or possibly a bigger problem with the current manual system, they agreed, but stated that they expected better results from a computerized system.

Four of the five members felt the system could be improved by providing more selective information to the

advisors for student advising and other uses. It is an interesting paradox that both members who stated that they liked the more complete real file, identified that one of the ways the system could be improved would be the removal of some unnecessary information from the reports given the advisors.

All of the interviewees commented very positively on the different ways information could be provided. Usually this was referred to as "lists".

Three of the members indicated that it would be very useful to include a section in the student file called "Future Plans" i.e. each student's plans for program specialization, courses to be taken, plans upon graduation.

6.1.1.3 Should DEACSS be continued - discontinued or continued in a modified form?

Staff members interviewed were unanimous that the system should be continued. One staff member suggested that "it would be better to keep less information and make certain that it is correct, than keep a lot of information which may contain errors."

The response to the question as to whether the system should be discontinued elicited the following sample answers:

1. "Dropping the project at this time would be premature. There is still a lot that can be done with it."
2. "Keep it, it is useful."

3. "I would be very disappointed to see it dropped - it is very useful."

6.1.1.4 What modifications could be made to the system?

Four of the staff members interviewed were of the opinion that the system should be expanded. The other offered that before any expansion should be attempted, the staff should discuss the implications of expanding the system. The following were suggested for expansion of DEACSS. In all cases this information was volunteered, not suggested by the interviewer.

1. Student File

   The following suggestions were received from one or more staff members.

   a. All members felt that a student's thesis title should be included in the student file. It was explained to each member that the thesis title, abstract and keywords were included when a person successfully defended his or her thesis. A number of the staff members suggested that the thesis topic be entered in the file as soon as the student proposed one.

   b. One staff member requested that a section entitled "Basis for Admission" be added. He suggested that this section include the relative weight placed by the admissions committee upon letters of reference, previous marks, and standardized tests (TOEFL, GRE).

   c. Two members indicated that they would like to see

implemented some method for obtaining follow up information on students. They suggested this information could be maintained in DEACSS.

2. Staff File

Three of the staff members identified that they would like to see the Staff File expanded to include data such as:

a. individual's publications.
b. committees on which staff members serve.
c. funded research staff members have undertaken.

One member expressed the opinion that no further information should be placed in the staff file without a full discussion with all staff members.

3. Course File

One member indicated that the following should be included and made available to students:

a. the course prerequisites and a description of each course should be included.
b. the text to be used and the requirements for completion should be stated for each course section.

4. Room Bookings

One staff member suggested that a timetable of all bookings for the Educational Administration Laboratory and the seminar rooms should be maintained in DEACSS.

5. Instruments and Questionnaires

One staff member suggested that research instruments and questionnaires used in thesis projects

should be indexed so that future students could access and evaluate them.

6. Miscellaneous comments and observations from academic staff:
   a. four members indicated that they felt DEACSS would be very useful in providing information for administrative decision making.
   b. one staff member keeps a file folder with all the student records output by DEACSS for his graduate students. He makes personal comments on the printed copy and states that he refers to this during the year when he needs to remind himself about part of a student's program or progress.

6.1.1.5 Summary of Academic Staff Interviews

The five staff members interviewed generally were positive toward DEACSS. The general concern was with the errors and missing information in the students' courses or marks. Most felt that while DEACSS did prove slightly useful in counselling students on their program, the real strength of DEACSS was in other areas - particularly in providing quick answers to unanticipated questions for administrative decision making.

6.1.2 Evaluation of DEACSS by Support Staff

In August 1980 before the first full test of DEACSS, the Department Secretary, who had been closely involved in the specification, design and implementation of DEACSS,

resigned. The Project Leader was then the only person who was completely familiar with the operation of DEACSS.

In September the administrative support staff (the new Department Secretary, the Administrative Assistant and the department's Programmer Analyst) were given a copy of the DEACSS User's Manual (Appendix A). One week later the Project Leader demonstrated the operation of DEACSS.

Although the Administrative Assistant and the Department Secretary indicated that they were apprehensive due to their unfamiliarity with the the use of the terminal and DEACSS itself, the support staff were pleased with the operation of DEACSS. The Programmer Analyst's evaluation was very positive. She indicated that the the system was designed in such a way that a clerk typist could easily use it. The attitude of the support staff appeared to be that DEACSS was now a part of the department operations and the suggestion by the Project Leader that it could be discontinued did not appear to be a desirable option. The major concern of the support staff was who was to be responsible for the actual interaction with DEACSS.

While this was to be a summative evaluation, the support staff were much more interested in improving the system than in criticizing it. The following enhancements to the system were suggested:

1. Data from the change of grade forms should be entered directly into DEACSS via the terminal rather than being keypunched. This would make the system much more

responsive to immediate change, and it would be less likely that the update information would not be entered through oversight.

2. Assistantship information should be entered into the system by the Administrative Assistant rather than the Department Secretary.
3. A yearly timetable of activities with respect to entering and modifying data in DEACSS should be prepared so that information in DEACSS would be current.
4. A list of students who still had incomplete grades in courses should be able to be generated.
5. A terminal totally dedicated to DEACSS and text processing should be obtained for the administration office.
6. At least one clerk-typist should be trained to use DEACSS so that the total responsibility for adding information to DEACSS would not reside with the Department Secretary.

6.1.3 Evaluation of DEACSS by Students

As was previously stated, at the time students met with their advisor for program counselling, they were given an individualized letter, a formatted copy of their DEACSS student file and a questionnaire. The cover letter explained with regards to the formatted copy of each student file, that "as this is the first time that this system has been used, there are bound to be errors or omissions (e.g. our course marks only go back to 1977, and marks for

Spring/Summer 1980 are not yet complete)." Each student was asked to check the information presented and:

1. correct any errors,
2. add any missing information (except marks prior to 1977 and for Spring/Summer 1980), and
3. indicate any information he or she would like removed from the record, explaining why it should be removed.

Students were asked to return the corrected files to the office with their questionnaire.

A number of part-time students either completed their re-registration without seeing their advisors or did not re-register for the fall session of 1980. After registration week, all questionnaires, letters and student files which had not been claimed by students were mailed to the last known address of the student. A total of 240 packages of materials were distributed to students. Of these packages, 128 questionnaires were returned, and 138 copies of student files were returned with corrections, additions or deletions.

The majority of the changes, additions or deletions to the student files requested by the students were corrections of typographical errors, correction or amplification of previous employment history information, or similar modifications. In four cases, the Project Leader judged that the information held in the DEACSS file was misleading or wrong to such an extent that if it were the only record of this student, an incorrect decision might have been made

regarding this student. In three of the four cases, courses were noted as incomplete which had been completed. In the fourth case, the work history information was completely incorrect. The high degree of accuracy of data in DEACSS can be credited to the Department Secretary who spent a great deal of time taking care that the information entered was copied as carefully as possible from the existing department file.

A few of the student records which were returned had comments written on them asking the relevance of the "Family Information" section. One student asked that this section be removed from her file.

The student questionnaire was generally intended to assess student attitudes towards computerized student records, not to act as an evaluation of DEACSS. Since the questionnaire was being completed immediately after the student was presented with a copy of his or her own student record as it was held in a computer system, it was felt that each individual's general attitude would be affected by this knowledge. Two main questions were included to assess the overall attitude of the student towards computerized student files. These questions, and the most common responses (those identified by over 5% of the respondents) to a subsequent open ended question for each of the main questions were:

1. As an administrator, do you feel that student records should be computerized?

   Yes 88% No 5% No Response 7%

   Why do you feel this way?

   Easy access to information..........................40%

   Efficiently store a great amount of information......40%

   Easy to update..................................20%

   Makes life easier..................................5%

2. As a student, do you feel that student records should be computerized?

   Yes 77% No 13% No Response 10%

   Why do you feel this way?

   Efficiently store a great amount of information......33%

   Easier access to information........................22%

   Privacy and access not guaranteed....................12%

   Records are more up to date..........................7%

   Increase speed and flexibility.......................6%

   Saves time in revising data..........................5%

   Simplification of record keeping.....................5%

As can be seen, student feelings change to some extent when they consider the use of a computerized system to store information on themselves, versus the prospect of accessing computerized student files as an administrator. The fact that there was only an 11% decrease in support when the students considered the use of their own records led to the

conclusion that the students generally accepted the storage of their own information on DEACSS.

A further indication that students were strongly opposed to having their personal information stored in a computer system would have been evidenced in comments written on either the questionnaire, on the corrected student files or made verbally to office and administrative staff. There were no negative comments about DEACSS written on either the questionnaire or the corrected student file, and no verbal comments were passed on to the Project Leader when office staff were asked. It is therefore assumed that the students accepted DEACSS.

6.1.4 Evaluation by Administrative Staff

In order to evaluate the usefulness of DEACSS to administrators, it was necessary to go beyond the Department of Educational Administration. There are six departments in the Faculty of Education at the U. of A. An individual appointment was made with each department chairperson to demonstrate DEACSS. Each chairperson was invited to ask his or her administrative support staff to attend this demonstration. If the department had a suitable terminal in its office, the demonstration was held there, otherwise it was held in the Project Leader's office.

Each demonstration,given by the Project Leader, was divided into four parts:

1. An explanation of DEACSS was given at which time samples

of DEACSS reports were presented. The rationale behind the study and how the Department of Educational Administration was benefiting from DEACSS was explained.

2. The protocol driven, interactive portion of DEACSS was demonstrated to show how data was entered or changed and how the predefined reports could be generated.
3. The capabilities of DEACSS to provide information on unanticipated queries was demonstrated. Each Department Chairperson was invited to ask for information which the Project Leader then attempted to find using native SPIRES commands. Since finding information on extemporaneous questions was beyond the capabilities of the predesigned SPIRES protocols, it was explained that in order to answer such unanticipated queries, someone in the department would need to learn some SPIRES commands.
4. Each administrator and his or her support staff were interviewed on their impressions of the usefulness of DEACSS.

The department chairpersons were very positive in their evaluation of DEACSS. The interactive prompting and predefined report capability was positively received, especially by the administrative support staff. Most of the chairpersons were more intrigued with the capabilities of the system when it was used in native SPIRES to provide information on previously unanticipated queries.

During three of the five interviews, department chairpersons asked if DEACSS could be made available to their departments. It was explained that the intention of the study was to develop a strategy which a department could follow in order to specify and implement a system such as DEACSS for itself. All of the department chairpersons requesting DEACSS commented that in its current state, it seemed to meet most of their existing requirements. It should be remembered that during the Predesign Analysis / System Specification stage the Project Leader interviewed most of the department chairpersons to show them the original design for the system and ask for improvements which would allow DEACSS to be generalizable to other departments. The response from the department chairpersons that DEACSS seemed to meet their existing requirements might indicate that previous suggestions received from these people were successfully incorporated into DEACSS.

The Department Chairman of Educational Administration and the Project Leader decided that any department requesting the DEACSS code should be given it. The only condition attached to the granting of the code was that proper credit be given to the author and the department.

Once DEACSS became operational, a number of agencies outside the Faculty of Education contacted the Project Leader about the findings and the possibility of using DEACSS. These included:

1. the Edmonton Public School Board,

2. the Faculty of Graduate Studies and Research at the University of Alberta,
3. the University of Victoria, and
4. Concordia College.

Edmonton Public School Board was interested mainly in the strategy and requested a copy of the thesis and the *DEACSS User's Manual*. The three latter organizations were all interested in building a student record file and were investigating the possibility of using DEACSS as a basis from which to build their own systems.

A demonstration similar to that held for the department chairpersons was presented to each of the last three groups. Subsequently, the Faculty of Graduate Studies and the University of Victoria have requested and received copies of the *DEACSS User's Manual* and the program code. A representative from Concordia College contacted the Project Leader some time after their demonstration and indicated that, while they were favourably impressed with DEACSS, they had not decided if they were going to place their student record files on a computer system and were still investigating alternatives.

6.1.5 Evaluation of DEACSS by the Chairman of the Department of Educational Administration

The ultimate user of DEACSS was to be the Chairman of the Department of Educational Administration. The Chairman was intimately involved in the project from its initial

stages. For the evaluation of DEACSS by the Chairman, the Project Leader set up a formal presentation similar to that given to all other administrators.

The Chairman stated that DEACSS had successfully achieved the original design parameters. He commented very positively upon interactive prompting capability within DEACSS which allowed easy addition, modification and reporting of data. The ability of DEACSS to generate output in a form which was compatible with *TEXTFORM, the University of Alberta's text formatting system, was deemed to be extremely useful.

Finally, the ability to generate information on previously unanticipated questions was obviously the capability which most impressed the Department Chairman. The usefulness of this capability was demonstrated during this interview when the Chairman mentioned a discussion in which he was currently involved, and for which some information was needed but could not be collected from the student files within the necessary time. The Project Leader retrieved the required information in approximately two minutes. Even though he had been involved in the specification and design of DEACSS, the Department Chairman was unaware that this information could be identified and made available so quickly and easily.

The Department Chairman was asked if there were any observations he had to make about the operation of DEACSS from his observations as the chairman of the department

actually implementing DEACSS. The following observations were forthcoming:

1. Since DEACSS was being used in parallel with the existing paper files, there was a problem insuring that DEACSS was always updated.
2. A timetable of recurring activities which utilized DEACSS should be built so that pertinent information would be entered at the appropriate time.
3. Even though most of the common requests for information from DEACSS were handled automatically by the use of the menu selection options, there were enough cases of the need for information not available through this method that someone in the department would need to gain familiarity with native SPIRES commands and the structure of the data base.
4. Since the information needs of the department change over time, someone would have to be retained to maintain the DEACSS data base and to build new SPIRES formats and protocols as they were required.

6.1.6 Evaluation of DEACSS by the Project Leader

The author was both the Project Leader and chief computer programmer for DEACSS. This meant that the author was intimately involved with all aspects of DEACSS, the design, implementation, and evaluation. It is, of course, difficult to give an unbiased evaluation of one's own project, but because of this unique involvement, the author is in a good position to provide an evaluation of both

DEACSS and the IDCSS strategy.

6.1.6.1 Problems with DEACSS

First, the final implemented version of DEACSS took much more time than had originally been planned. This was due to many things:

1. systems problems in SPIRES caused delays.
2. the entry of data was handled completely by the Department Secretary during slack time, again causing delays.
3. some modifications made to the initial version of DEACSS required thought over an extended time period before a solution which was compatible with the design parameters was found. This thought time could not be estimated accurately for inclusion in Chapter five, since it was not a matter of being able to spend "X" hours at a single time to find a solution - rather it was a case of occasionally thinking about the problem over a long period of time and allowing possible solutions to emerge and evolve until, eventually, a "best" solution became apparent.
4. the initial plan was to simply build a graduate student record file. During the Predesign Analysis it became apparent that a good graduate student record file would be inefficient and ineffective without course and staff files. DEACSS became a much more comprehensive system than originally was planned.

5. It took longer than originally anticipated to develop the interactive routines to insure that a naive user would be able to use the system.

Second, the Summative Evaluation stage was undertaken at a very inappropriate time. The Department Secretary had been replaced just prior to the first full scale trial of the system. In addition it would have been better to wait for a year until some of the problems which were bound to appear in this first complete test of the system had been corrected.

Third, as the project progressed, there were occasions when the Project Leader found himself putting greater value on suggestions received from people who were generally supportive of the project, while putting less value on suggestions received from people who were nonsupportive. This is not an approach which leads to optimum satisfaction with the system by its users.

6.1.6.2 Positive Features of DEACSS

Generally DEACSS can be regarded as a success. It meets the current information needs of the department and has been demonstrated to be easily modifiable. The Chairman of the Department of Educational Administration, the prime user, is satisfied with its performance. The number of outside requests for DEACSS also supports the usefulness of the system.

6.2 An Evaluation of the IDCSS Strategy

Since it has been used only once, for the implementation of DEACSS, it is not possible to give a full evaluation of the design strategy. However, from the point of view of this single application, the strategy seems to work reasonably well.

One interesting possibility is that the strategy may work too well. The author attempted to design and build a very generalized system. Many people from outside the actual department for which the system was to be implemented were consulted. When DEACSS was completed many of these people were more interested in obtaining DEACSS itself rather than using the strategy to design and implement their own system.

Since the strategy was used only a single time, there are many possibilities of idiosyncratic behavior of which others wishing to utilize the strategy should be aware. The Project Leader and the Department Chairman have the following words of caution which arise from this specific application.

1. The Project Leader, the System Designer and the chief computer programmer were all the same person, the author. Because of the unique position of the author, that is as a staff member who has been a professional computer analyst, a great deal of communication time may have been saved in that, in this case, the System Designer, computer programmer and the Project Leader

usually understood each other.

2. At times during the project, some people were identified as "resistors" when this may have been inappropriate. While it may seem reasonable to classify people as "adopters" or "resistors" this should not be done. Treating a person as a "resistor" may turn that person into one.
3. It is difficult to get staff involved in the implementation stage due to their lack of knowledge of computer technology.
4. Budget for operating personnel from the beginning of the project. DEACSS was implemented using slack time from existing operating personnel. This caused time delays and at times caused uneasiness for the operating personnel as they felt that something needed to be done on DEACSS, but their regular tasks also demanded attention.
5. Personal experience would indicate that the possibility of success of the implementation may be directly proportional to the funds allotted to the project.

6.3 Chapter Summary

The design and implementation of DEACSS was a success. Academic staff, support staff and administrators are all generally pleased with the implementation. Students, while not asked to evaluate DEACSS directly, provided information which indicated acceptance of DEACSS. This positive

evaluation came despite the fact that the Summative Evaluation was undertaken earlier than it should have been.

The strategy used to implement DEACSS worked successfully on this occasion. Since this application was a case study, one should be very hesitant before applying the strategy without attempting to see if it has a reasonable chance for success given the individual differences in organizations.

7. SUMMARY AND IMPLICATIONS

This study was undertaken to assess the possibility of developing a strategy which could be used by an instructional department to design and implement a computer support system which would meet that department's needs. The intention was to develop a strategy which would be generalizable to many instructional departments and to test this strategy by implementing a full scale instructional department computer support system (IDCSS) for one department - the Department of Educational Administration at the University of Alberta. Major concerns during the development of the strategy were that the IDCSS should:

1. be economical to develop,
2. be sufficient to meet the current information needs of the department, and
3. be easily modifiable and extensible in order to meet the constantly changing and expanding information needs of the department.

The purpose of this study was met in that:

1. A strategy for the design and implementation of an instructional department computer support system was developed.
2. This strategy was tested by using it to implement a fairly large, in both scale and scope, computer support system - the Department of Educational Administration Computer Support System (DEACSS).

3. A set of cost figures for the implementation of DEACSS were reported. These costs were subdivided into different task categories, were subtotalled for each of the stages, and finally were totalled for the complete project.
4. A number of users and potential users were asked to evaluate DEACSS to see how well it met their needs.
5. The strategy used to implement DEACSS was evaluated.

In order to place the study in perspective, an extensive review of the literature was first presented. This included an examination of the way computer information systems had developed, of the way data base management systems and management information systems worked, a discussion of some of the problems and successes of MIS's, particularly in the field of education, and a discussion of decision support systems.

The results of this review of the literature, combined with the prerequisites of economy, sufficiency and extensibility led to the selection of a MIS approach to the study. The examination of change theory literature involving computer systems in organizations, along with the author's personal experience indicated that the users, in this case department staff members, both individually and collectively, would have to feel involved in the design of the system if the system were to be accepted. The development of the plan to use a MIS approach combined with concerns about gaining acceptance in the organization led to

the general strategy which was proposed in Chapter two. This general strategy was composed of seven stages, some of which were to be undertaken simultaneously. These stages were:

1. Perception of the Need for a Computer Support System,
2. Predesign Analysis / System Specification,
3. Establish Positive Department Climate (undertaken simultaneously with Stage 2),
4. Design System,
5. Implement System,
6. Formative Evaluation (undertaken simultaneously with Stage 5), and
7. Summative Evaluation.

The proposed strategy was tested by using it to specify, design, implement and evaluate DEACSS. The SPIRES DBMS was chosen as the vehicle for implementation because of its availability and many useful features. Throughout the study a record of person hours and computer time was maintained. This record was intended to give prospective users of the strategy some feeling for the order of magnitude of time and costs involved.

While the department had perceived a need for a IDCSS for some time, the study was formally undertaken in February 1979. In order to make the system as easy as possible to use, it was decided that, wherever DEACSS was to be used by personnel who were not employed in computer oriented jobs, the system should lead the user to the required result by means of prompts and menu selection. While this meant a

great deal of extra design and programming, it reduced staff training to a minimum. It also hid most of the complexity of DEACSS from the user.

Once the department began to use DEACSS, a number of enhancements and modifications were requested. These resulted in many minor and a few major revisions to DEACSS. While the specific changes which were requested were unknown at the early stages of the design of DEACSS, the probability of requests was anticipated, hence the system was built to accommodate changes.

The final implementation of DEACSS consisted of three inter-related SPIRES subfiles:

1. a graduate student record file containing personal, professional, and course information for students enrolled as graduate students in the Department of Educational Administration. Any graduate or undergraduate student who had taken a course from the department, even if they were not enrolled in the department, had a minimal entry in this file which consisted of name, student identification number, course and mark.
2. a course file containing information on courses offered by the department. One aspect of particular interest in this file is the year, session and section structure which contains a pointer to the student file for each student enrolled in the course.
3. a staff file containing basic information on any person

who:

a. was appointed to the academic staff of the department,

b. had taught a course for the department, or

c. had served on any thesis committee in the department.

In September 1980 a full scale trial and summative evaluation of DEACSS was undertaken. It was understood by both the Department Chairman and the Project Leader that the summative evaluation was premature, but circumstances dictated that the evaluation be done. Students, support staff, academic staff and administrators assisted in the evaluation of DEACSS.

A sample of academic staff members in the department were interviewed after they had used the printed student records produced by DEACSS for counselling graduate students. These staff members were unanimous in their judgement that DEACSS should be continued. A few weaknesses were identified, outstanding among them was that not all courses which a student had taken were reported.[15] These staff members indicated acceptance of DEACSS as a department system by referring to it as "our" system or "the department" system rather than as "your" (the author's) system.

----

[15] Only courses taken in the department after January 1977 were entered to DEACSS at that time - a serious limitation for counselling purposes.

Students did not know they were assisting in the evaluation of DEACSS. They were given a copy of their own formatted student file and a questionnaire which addressed the kind of data found in that file and their attitudes towards having their own information stored in a computer file. In addition, they were asked to correct their own student record and to return the corrected copy to the department office. This latter procedure acted as a measure of the accuracy and completeness of the information in DEACSS. Approximately 50% of the students who were registered in the department returned a questionnaire. Almost all of these returned their copy of the student record with corrections or additions. Most of these corrections were for spelling errors or employment clarification. Some students included much more complete information than was available in the existing department files. Four students made corrections to information in their file which could, in isolation, have misled individuals reading the student file.

DEACSS was demonstrated to the Administrative support staff in the department who would use DEACSS on a daily basis. Their general attitude was positive, with any apprehension relating only to a general unfamiliarity with computer systems and hardware.

DEACSS was demonstrated to all Faculty of Education department chairpersons, and many of their administrative staff. These people were very positive towards DEACSS. While

the protocol driven interactive system was well received, the capabilities of generating answers to previously unanticipated questions seemed to elicit the strongest positive responses, especially from the chairpersons themselves. It was explained that in order to use DEACSS to obtain information on unanticipated queries, the user would have to learn some SPIRES commands. This did not seem to alarm the chairpersons, although some of them indicated that they would have someone else learn SPIRES.

Of the five department chairpersons to whom DEACSS was demonstrated (excluding the Chairman of Educational Administration) three asked if they could have copies of DEACSS to use DEACSS in their own department. It was explained that the intention of the research was not to have DEACSS generalizable to all departments, but that the strategy used to implement DEACSS would be generalizable. The department chairpersons indicated that DEACSS itself seemed to have most, if not all, the features they could forsee, and they wished to investigate the implementation of a modified version of DEACSS.

Three groups outside the Faculty of Education also asked for a demonstration of DEACSS. After the demonstrations, both the Faculty of Graduate Studies and Research at the University of Alberta and the University of Victoria asked for and received copies of the DEACSS code so they could implement their own systems.

The evaluation of the generalized strategy used can only be made by implication from the single case in which it was used - the design and implementation of DEACSS. DEACSS appears to have been a success. More work certainly needs to be done to rectify a few problems or add features identified during the Summative Evaluation stage, but the system seems to be successfully meeting the information needs of the Department of Educational Administration.

The total costs for the complete test of the strategy in implementing DEACSS were $4740 in computer costs, 539.5 person hours for the Project Leader, 492.5 person hours for computer analysts and 422 person hours for clerical staff. While all of the clerical staff time was provided by existing staff in slack time, anyone thinking of implementing a system such as DEACSS should consider the above as a real cost.

Academic staff were consulted for 30.5 person hours, while the department chairman provided 17 person hours. These might be considered as "no cost" services, since these people were already available. People outside the department provided 50.5 person hours, all of which were provided at no charge.

As the Project Leader, the author would judge both the general strategy and DEACSS itself as successful.

7.1 Implications

The creation of a strategy which enables an instructional department to implement its own computer support system has many positive implications:

1. An instructional department will be able to store and retrieve precisely the information it needs. It will not be dependent upon the central administration for what can be stored, in what form it can be retrieved, or when the service can be supplied.
2. Many trivial but time consuming tasks, such as the preparation of lists of students in certain programs, can be performed with a single command using such a system. This frees support staff for other work.
3. Information which would previously have been very time consuming to obtain can now be provided almost instantaneously.
4. A well designed IDCSS will provide information in an effective and efficient manner, hence allowing the administrator to spend time making decisions rather than collecting information.

There can also be negative implications:

1. A department cannot normally provide the stringent checks on the accuracy of data that the central administration can.
2. Central administration may perceive an IDCSS as a threat to its control.

This research has demonstrated that it is possible for an instructional department to develop its own IDCSS to its own specifications. With the strategy developed in this research, and access to a powerful DBMS such as SPIRES,any instructional department whose members feel that a CSS would benefit them should be able to successfully implement its own instructional department computer support system.

7.2 Suggestions for Future Research

This research has opened the door to a great deal of possible future research. Some questions which might be pursued follow.

1. Does DEACSS provide better quality information?
2. What are the attitudes of students and staff towards a system such as DEACSS?
3. Does and if so, how does the decision making in the department change after the implementation of DEACSS?
4. Can other instructional departments implement the strategy with the same success experienced in implementing DEACSS?
5. Now that information on graduated students is easily available, can follow-up studies on students be undertaken?
6. Now that information on graduated students is easily available, can longitudinal studies on students be undertaken?

7. Now that information about part-time students is more easily available, can the department find ways to include these part-time students in department activities, possibly with the result of improving the percentage of part-time students completing the program?

7.3 Final Summary

A strategy to develop an IDCSS was developed in this thesis. The strategy was followed to implement DEACSS. A great deal of work remains to be done to DEACSS to bring it up to the standards which the author would like, but it does generally meet the needs of the Department of Educational Administration for an instructional department computer support system.

The implementation of DEACSS was a success, hence the strategy worked for this single application. The strategy needs to be applied in other departments before it can be rated a general success, but it is hoped that others will find it useful in implementing an IDCSS for their own use.

SELECTED REFERENCES

Alter, S.L. *Decision Support Systems: Current Practice and Continuing Challenges*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1980.

Antworth. This was Your First Effort at TOP-DOWN Design, Wasn't It. *Datamation*, November 1980, *26(11)*, 182.

Aron, J.D., Information Systems in Perspective, *Computing Surveys*, December 1969, *I*, No. 4, p. 213.

Caffrey, J.G. The Innovation Matrix. Paper presented at *The Institute for Government and Public Affairs Conference on Educational Innovations*, UCLA Lake Arrowhead Center, 1965, p. 14. As quoted in Havelock, R.G. *The Change Agent's Guide to Innovation in Education*. p. 111.

Clemson, Barry, Harnessing the Computer in Educational Management. *Journal of Educational Administration*, July 1980, *XVIII(1)*, 98 - 113.

Cohen, L.J. *Data Base Management Systems*. Wellesley, Massachusetts, Q.E.D. Information Sciences, Inc., 1978.

Collins, J.F., Feeney, G.J. and Gosden, J. Calling a Spade a Spade ... A Chat with MIS Executives. *Datamation*, November 1979, *25(13)*, 59-62.

CUMREC '78 - Proceedings of the 23rd Annual College and University Machine Records Conference. London: 1978.

CUMREC '80 - Proceedings of the 25th Annual College and University Machine Records Conference. Salt Lake City, Utah: 1980.

Davis, A. Personal Conversations 1975-1980.

Digital Equipment Corporation. *PDP 11 Systems and Options Summary*. Maynard, Massachusetts: Digital Equipment Corporation, 1979, 91 pp.

Gruhn, A.M., and Hohl, A.C. A Research Perspective on Computer-Assisted Office Work *IBM Systems Journal*, Vol. 18, #3, 1979, pp. 432-456.

Hansen, T., Klassen, D. and Lindsay, J. The Impact of Computer-Based Information Systems Upon School and School District Administration. *AEDS Journal*, Vol. 12, #1, Fall, 1978, pp. 1-10.

Havelock, R.G. *The Change Agent's Guide to Innovation in Education*. Englewood Cliffs, N.J.: Educational Technology Publications, 1973, 279 pp.

Holland,R.H., DBMS: Developing User Views, *Datamation*, February 1980, *26*, No. 2, p. 141-144.

Holsapple, C. *A Primer on Data Base Management Systems.* Lafayette, In.: Micro Data Base Systemns, Inc., 1980, 54 pp.

Hussain, K.M. *Development of Information Systems for Education*. Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 1973.

Jackson, G.R. *SPIRES/370 File Definition*. Edmonton: Department of Computing Services, The University of Alberta, 1977.

Jackson, G.R. and Davis, A. *SPIRES User's Guide*. Edmonton: Department of Computing Services, The University of Alberta, 1976.

James, R. Personal Conversations 1975-1980.

Keen, P.G.W. and Scott Morton, M.S. *Decision Support Systems: An Organizational Perspective*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1978.

Keen, P.G. and Wagner, G.R. DSS: An Executive Mind-Support System. *Datamation*, November 1979, *25(12)*, 117-122.

Korenthal, J.E., Data Base Concepts. *CUNY/UCC Communications*, 1978, *4(1)*, 2-3, *4(2)*, 10-11, *4(3)*, 22-23, *4(5)*, 38-42.

Kropf, John. Student Information System for a Small University. In *CUMREC '80 - Proceedings of the 25th Annual College and University Machine Records Conference*. Salt Lake City, Utah: 1980, 61-73.

Lewin, K. Group Decision and Social Change. In T.M. Newcomb and E.L. Hartley (Eds.), *Readings in Social Psychology*. New York: Holt, 1952.

Micro-Ap. *Selector III-C2 Information Management System: Operations Manual*. San Ramon, Ca.: Micro-Ap, 1979, 79 pp.

Montgomerie, T.C. *DEACSS User's Manual*. Edmonton, Alberta, Department of Educational Administration, University of Alberta, 1980.

Montgomerie, T.C. *DEACSS Technical Manual*. Edmonton, Alberta, Department of Educational Administration, University of Alberta, 1981.

Montgomerie, T.C. Attitudes of Educational Administrations Towards Automated Student Record Systems. In preparation 1981.

Oettinger, A.G. and Marks, S. *Run, Computer, Run: The Mythology of Educational Innovation*. Cambridge, Massachusetts: Harvard University Press, 1969, 302 pp.

Picciano, A.G., Use of a Statistical Package for Management Data Reporting. In *CUMREC '78 - Proceedings of the 23rd Annual College and University Machine Records Conference*. London: 1978, 259-272.

Plice, S.J. Administrative Information Systems - Preparing for the Eighties. In *CUMREC '80 - Proceedings of the 25th Annual College and University Machine Records Conference*. Salt Lake City, Utah: 1980, 283-289.

Richard, H.G., Richard, V.H. A Talon for the Phoenix. In *CUMREC '78 - Proceedings of the 23rd Annual College and University Machine Records Conference*. London: 1978, 117-128.

Roland, J. The Microelectronic Revolution: How Intelligence on a Chip Will Change Our Lives. *The Futurist*, April, 1979, pp. 81-90.

Schein, E.H. Management Development as a Process of Influence. *Industrial Management Review*. 2(2), Spring 1961, pp. 59-77.

Scheinbien, R. Computer Selection - an invited presentation in a the course *EDADM 506 - Computing Concepts in Educational Administration* at the Univsity of Alberta on August 13, 1980.

Schwartz, M.H. MIS Planning. *Datamation*, September 1970, *16(10)*, 28-31.

Scriven, M. The Methodology of Evaluation. In R.E. Stake (Ed.), *Perspectives of Curriculum Evaluation*, American Educational Research Association Monograph Series on Curriculum Evaluation, No. 1. Chicago: Rand McNally, 1967. Pp. 39-83.

Seger, J.E. and McKinney, J. *Department of Educational Administration Student Information Brochure*. Edmonton, Alberta, Department of Educational Administration, University of Alberta, 1979.

Senda, R. Personal Conversations 1975-1980.

Senda, R. Data Base Management Systems and Management Information Systems - an invited presentation in the course *EDADM 506 - Computing Concepts in Educational Administration* at the Univsity of Alberta on August 7, 1980.

Senda, R. Stanford Public Information Retrieval System (SPIRES). In *Proceedings of the Ninth Annual Meeting of the American Society for Information Science - Western Canada Chapter*. Banff, Alberta: 1977, 93-100.

Slotnick, D.L. & Slotnick, J.K. *Computers: Their Structure, Use and Influence*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1979.

Solomon, M.B. Economies of Scale and the IBM/360. *Communications of the ACM*, June 1966, *9(6)*, 435-440.

Tsichritzis, D.C., Lochovsky, F.H. Designing the Data Base. *Datamation*, August 1978, *24(8)*, 147-151.

University of Alberta, Department of Educational Administration, *Minutes of the Department Meeting*, Sept. 5, 1980.

Weizenbaum, J. *Computer Power and Human Reason*. San Francisco, California: W.H. Freeman and Company, 1976.

Yourdan, E., Constantine, L.L. *Structured Design*. New York, New York: Yourdon Inc., 1975.

Zand, D.E. & Sorensen, R.E. Theory of Change and the Effective Use of Management Science. *Administrative Science Quarterly*, December 1975, *20*, 532-545.

APPENDIX A - DEACSS USER MANUAL

DEPARTMENT OF EDUCATIONAL ADMINISTRATION

COMPUTER SUPPORT SYSTEM

(DEACSS)

USERS MANUAL

T. Craig Montgomerie

October, 1980

Abstract

This document describes the operation of DEACSS, The *D*epartment of *E*ducationnal *A*dministration *C*omputer *S*upport *S*ystem. DEACSS was designed to be an evolutionary, state of the art, Computer Support System aimed at supporting the both the decision making and the clerical needs of an instructional department at the University of Alberta.

This document has been written in such a way that the casual user need only read the first two sections to use DEACSS. Section 1 introduces DEACSS and explains its capabilities, while Section 2 explains Interactive DEACSS - a simple to use, menu driven program which encompasses most of the normally required functions of DEACSS.

The advanced user should read the complete document. The author assumes that the advanced user has read the manuals on SPIRES and MTS cited in the "Technical References" section, hence normal MTS and SPIRES terminology will be used in the advanced sections without explanation.

Even though the manual has been written to be understood by a novice user, some features may be a little unclear. In order to show the user how the system works, examples have been liberally spotted throughout the document to assist wherever it is felt they may be necessary.

Table of Contents

Chapter Page

1. An Introduction to DEACSS

This document describes the operation of DEACSS, The *D*epartment of *E*ducational *A*dministration *C*omputer *S*upport *S*ystem. DEACSS was designed to be an evolutionary, state of the art, Computer Support System aimed at supporting the both the Decision Support and the clerical needs of an instructional department at the University of Alberta.

DEACSS is written in SPIRES, the *S*tanford *P*ublic *I*nformation *RE*trieval *S*ystem and runs on the General Computing Facility operated by the Department of Computing Services at the University of Alberta. While the system is built so that the casual user needs no knowledge of SPIRES and little knowledge of MTS (*M*ichigan *T*erminal *S*ystem - the operating system under which SPIRES operates), the advanced user would be advised to have read the references in the Technical References Section.

Currently DEACSS contains three SPIRES subfiles:

1. a student file (GRAD RECORDS)
2. a staff file (STAFF FILE)
3. a course file (COURSE FILE)

These files are linked to each other so that, for instance, the "Advisor" entry in the student file points to the appropriate staff record. Correspondingly, each staff record has elements which point to the student file record of each student that staff member advises.

The actual information stored in each of these files is given in Appendix D - A Summary Data Element Dictionary.

DEACSS has two levels of operation:

1. The most commonly required facilities have been built into a "Menu Driven" protocol which has been called "Interactive DEACSS". This enables the casual user to access most of the information, reports, etc. without any knowledge of SPIRES or the file structure.
2. A knowledgable user can enter native SPIRES, and can therefore do anything SPIRES allows.

2. Interactive DEACSS

This section describes the features of DEACSS which have been implemented in an easy to use, menu driven fashion. This facility has been called Interactive DEACSS. Standard reports are generated by using Interactive DEACSS and the addition and modification of data in DEACSS should, wherever possible, be made through the Interactive DEACSS as error checking is performed here which is not available when accessing DEACSS via Native Spires.

Any features not available in Interactive DEACSS can be added by the author, or by a SPIRES consultant.

2.1 Accessing Interactive DEACSS

DEACSS is implemented on the General Computing Facility at the University of Alberta. As such, it can be accessed through DATAPAC[1] from almost anywhere in the world! Currently, for obvious security reasons, DEACSS is available only to a single Computing Services Identification (SPIS). In order to fully utilize the system, two pieces of equipment are required:

1. a Courier, Andersen Jacobson 510, IBM 3277 terminal, or other terminal equipped with visual editing facilities, and
2. An Andersen Jacobson 832 Printing terminal.

----

[1]DATAPAC is the tradename for packet switched computer data communications network which is operated by the Trans-Canada Telephone System. Datapac has the ability to interface to other data communications networks operating throughout the world.

The procedure for accessing Interactive DEACSS is:

1. Sign on to the Computer:

    $SIGNON SPIS

2. Enter the Password:

    The password will be made available to a valid user by the author.

3. Enter the interactive program:

    $SOURCE CONTROL

At this point the system will print out any messages about recent modifications to DEACSS and will generate the "Main Menu" of options. Figure 1 shows a copy of the interaction with MTS and Interactive DEACSS up to this point.

In this and all future examples of interactions between the computer and the user, anything printed by the computer will be printed in normal type face. Any command or response by the user will be given in **bold type face**.

2.2 Adding or Modifying Student Records

Once you have chosen this option, Interactive DEACSS will ask you to enter the surname or the University of Alberta Student Identification Number for the student whose record you wish to add or modify. Since the U. of A. ID. is unique and immediately identifies the student, this is the preferable choice. If the ID is not known and the surname is entered, DEACSS will respond by giving a list of all the students with that surname and their ID number. You will then be asked if the student you wish is in the

Figure 1

ACCESSING INTERACTIVE DEACSS

```
# SIGNON SPIS
# PASSWORD?
?
# TERM,PRIME,INTERNAL/TEACHING,RESEARCH
# **LAST SIGNON WAS: 16:24:10
#  USER "SPIS" SIGNED ON AT  16:25:46 ON THU JUL 10/80
# source control
# $SET ECHO=OFF
> *** MESSAGES
>
>    Messages to the Departmental Secretary
>    as to modifications or additions to
>    Interactive DEACSS appear here.
>
Spires active file set to: -SAF
-Spires 6.0
-Tracing on, 'EXPLAIN TRACING' for details
-?) SET NOECHO
  *
  *
  *
  *                  DEPARTMENT OF EDUCATIONAL ADMINISTRATION
  *                          COMPUTER SUPPORT SYSTEM
  *
  *   MAIN MENU
  *   **** *****
  *
  *   Currently, you have the following options:
  *   1) Add or modify individual student records.
  *   2) Add a new staff record.
  *   3) Enter a group of students requesting
  *      assistanceships.
  *   4) Generate a Report.
  *   5) Generate *TEXTFORM compatible output.
  *  98) Enter Native Spires Mode (do not processthe
  *      file).
  *  99) Return to MTS (do not process the file).
  *
  *      To terminate the run normally, hit "ATTN" or "PA1".
  *

:Please enter the number of an option:
```

list and if so to enter the appropriate ID number.

Once you and the system have reached agreement as to whether or not the student exists, you will be transferred to one of two possible states.

1. the addition of a new student record, or,
2. the modification of an existing student record.

2.2.1 Addition of a New Student Record

In this state, the program will prompt you to enter all the information necessary for a new student record. This information is contained on:

1. the Department of Graduate Studies Application for Enrollment Form.
2. the Department of Educational Administration Application Form.

A sample interaction for an imaginary student (Alan A. Aardvark) is shown in Figure 2. An example of the record built by the interaction in Figure 2 is shown in Figure 3.

2.2.2 Modification of an Existing Student Record

If a record already exists for a student, Interactive DEACSS will generate the menu of options shown in Figure 4. Each of these options is discussed below.

2.2.2.1 Change the ID number for the student

The ID number of the student is the most important element in a student record. In SPIRES terms, it is the "KEY" to the record, that is, the element which is guaranteed to be unique for

Figure 2

ADDING A NEW STUDENT TO DEACSS

```
    . . .
    . . .
    . . .
:Please enter the number of an option: 1
    *
    *       DEPARTMENT OF EDUCATIONAL ADMINISTRATION
    *              COMPUTER SUPPORT SYSTEM
    *
    *  Individual Student Addition and Modification Section
    *  ********** ******* ******** *** ************ *******
    *
:Enter Student Surname or ID: Aardvark
    * This must be a student for which we don't have a record.
:Should I create a new record for this student? yes
:Do you know the U of A Student ID? yes
:Please enter the Student ID. 808080
    * I am creating a new record for this student.
    *
:Please enter the legal name (Last, First Second, Title)
                              Aardvark, Alan A., Mr.
:Enter the name by which the student wishes to be addressed Alan
:Please enter the Social Insurance Number: 999 999 999
:Enter the student address 99 Llama Road
:Enter the city Edmonton
:Province/State? Alberta
:Country? Canada
:Postal/Zip Code? T6G OM5
:Phone? 499-9999
    *
    * Enter Address type:
    *    1 - Permanent Address
    *    2 - University Address
    *    3 - Business Address
:Please Enter 1, 2, or 3: 1
:Do you have a Dept. Application form for this student yes
    *
    *          Department of Educational Administration
    *                  Application for Admission
    *
:Date of Birth? Sept. 14, 1945
:Place of Birth? Hoboken, New York
    *
    * Marital Status can be:
    * 1 - Single
    * 2 - Married
    * 3 - Seperated
    * 4 - Widowed
    * 5 - Divorced
    * 6 - Unknown
:Please enter the Marital status of the student: 2
```

```
:Enter the spouses name (Last, First, Preferred Reference):
                        Aardvark, Agnes, Mrs.
:Does the student have any dependents? yes
:How many dependents? 1
:Do you know the names of the dependents? yes
:Enter a dependent's name (Null = Unknown, Attn = No more):
                          Aardvark, Anthony, Mr.
:Enter the birthdate of this dependent: June 6, 1979
:What is the relationship? son
:Enter a dependent's name (Null = Unknown, Attn = No more):
    ***Attn
    *
    * ACADEMIC EXPERIENCE
    *
:Enter a degree (Just return if no more): B.Ed.
:Specialization? (Nursing, Elementary Ed., etc.): Elementary
:University? University of Alberta
:Graduation Year? 1967
:Enter a degree (Just return if no more):
    * ACADEMIC AWARDS AND SCHOLARSHIPS
:Enter the name of an award (ATTN if no more):
                             Queen Elizabeth Scholarship
:What is the nature of the Award (Bursary, Scholarship:
                                 Scholarship
:Where was the award obtained: University of Alberta
:When was the award applied for (if not yet received):
:Enter period award held: 1967
:Enter the amount of the award : $500.00
    *
    * The Current status of the award can be:
    * 1 - Pending
    * 2 - Granted
    * 3 - Renewed
    * 4 - Completed
    * 5 - Refused
    * 6 - Unknown
:Please enter the Current status of this award: 4
:Enter the name of an award (ATTN if no more):
    ***Attn
    *
    * CERTIFICATES IN EDUCATION
    *
:Enter the Certificate granting Agency (ATTN if no more): ASTD
:What is the number of the Certificate 999999
:Enter the date of issue: Sept. 1, 1978
:Enter the duration of the certificate (Permanent or Final Year):
                                       Permanent
:Enter the Certificate granting Agency (ATTN if no more):
    ***Attn
    *
    * WORK - EXPERIENCE
    *
:Enter the name of the Employer (ATTN if no more): E.P.S.B.
:Enter the Employer address 10010-107A Ave.
```

```
:Enter the city Edmonton
:Province/State? Alberta
:Country? Canada
:Postal/Zip Code? T4G 0N0
:Phone? 429-5621
:Enter the Job Title: Teacher
:Enter the dates for which this job was held: 1974 to present
    *
    * Enter a description of the job (A null line will end it).
:: Teaching
::
:Was this job in the education field? yes
    * Enter the Educational Specialization (Counselling,
      Adminstration, Teaching, etc):
:: Teaching
:Enter the type of student: Grade 1-3
:Was another position held with this employer? no
:Enter the name of the Employer (ATTN if no more):
    ***Attn
    * Enter the title of a publication (Attn if no more)
:: The Engineer as a Teacher
:Enter the date of Publication: July 1979
:Enter the name of a co-author: Smith, W.P.
:Enter the name of a co-author:
:Journal or Publisher: The Education Digested
:Enter the number of pages (Book) or page range (Journal): 23-43
    * Enter the title of a publication (Attn if no more)
::
   ***Attn
   *
   *  At this point, you have the following options:
   *
   *  1 - Finish with this student - go back to the control
          program.
   *  2 - Enter minimal current registration information
:Please enter the number of an option 2
    *
    * Possible enrollments in the Department of Educational
    * Administration are:
    *  1 - Diploma
    *  2 - M. Ed.
    *  3 - Ph. D.
    *  ATTN to Abort this change
    *
:Please enter the number of an option: 2
    *
    * The possible programs for Masters students are:
    * 1 - Thesis
    * 2 - Non-Thesis
    * 3 - Administration Development Program
    * 4 - Teaching Skills Improvement Program
    * 5 - Unknown
:Please enter the number of the program type 1
    *
```

```
    * The current status of the student in the Masters
    * program is:
    * 1 - Application Pending
    * 2 - Application Refused
    * 3 - Qualifying Graduate Student
    * 4 - Candidate for Masters
    * 5 - Granted Masters Degree
:Please enter the current status of this student 4
    *
    *
    * Please enter the official date for the action.
:(Just "Enter" for today, "ATTN" for unknown).
    * The current enrollment status is:
    * 1 - Full Time
    * 2 - Part Time
    * 3 - Not Yet Enrolled
    * 4 - Registration Lapsed
    * 5 - Withdrew
    * 6 - Convocated
:Please type 1 or 2 1
    *
    *
    * Please enter the official date for the action.
:(Just "Enter" for today, "ATTN" for unknown).
    * Please enter the advisor's last name:
:(Just "Enter" if Advisor is unknown). Seger
    * One staff member has been found with:

     STAFF-ID = 24;
     STAFF-NAME = Seger, J.E., Dr.;
     DEPT = Department of Educational Administration;
:Is this the staff member you want? yes
    * Please enter the date the Advisor was appointed.
:(Just enter for today, ATTN for unknown).
    * Please enter a committee member's last name:
:(Just "Enter" if no more members)
:Has this student requested assistance? yes
:Please enter the date the Assistanceship would begin:
                                  Sept. 1, 1980
:Please enter the date the Assistanceship would end:
                                  April 30, 1981
:Has a decision be made on granting this assistanceship? yes
:Are there any special comments to be entered? yes
    * Enter the Comment (ATTN when finished)
: This is a dummy student
:
  ***Attn
:Has this student applied for Advanced Credit for courses? yes
    * Please use the:
    *     *******************************************
    *     * Faculty of Graduate Studies and Research *
    *     *                                         *
    *     *    RECOMMENDATION FOR ADVANCED CREDIT   *
    *     *******************************************
```

```
    *
    * We will follow this form and handle one course at a time.
    *
:What is the date on this form (NULL -Today, ATTN - Unknown)
    *
    * Enter the Degree for which Advanced Credit is to be
    * applied:
    *  1) Diploma
    *  2) Masters
    *  3) Ph.D.
: 2
:Was the first course taken at the University of Alberta? yes
    *       ********************************
    *       *   U of A Course Information   *
    *       ********************************
:Please enter the Course Abbreviation and Number: EDPSY502
:Please enter the Year in which the course was taken: 1976
:Please enter the Session in which the course was taken: Spring
:Please enter the Course Section: A3
:Please enter the Faculty in which the course was taken:
                                                  Education
:Please enter the Category of the student at the time the
 course was taken: Special Student
:Please enter the Course Mark: 9
:Are there any special comments for this course? yes
    * Enter a comment (ATTN when finished)
: This course has been granted equivalent to EDADM511/512
:
  ***Attn
:Are there any other courses to enter? no
:Do you want to work on another student record? no
    *
    * OK - I'll return you to the Supervisor Program.
    *
    *   Currently, you have the following options:
         . . .
         . . .
         . . .
```

Figure 3

AN EXAMPLE STUDENT RECORD

```
IDNO = 808080;
LEGAL-NAME = Aardvark, Alan A., Mr.;
PREFERRED-NAME = Alan;
MODDATE = July  10, 1980;
DEPARTMENT = Educational Administration;
S-ADDR-TYPE = 1;
  S-STREET = 999 Llama Road;
  S-CITY = Edmonton;
  S-PROVINCE = Alberta;
  S-COUNTRY = Canada;
  S-POSTAL-CODE = T6G 0V0;
  S-PHONE = 499-9999;
  S-VERIFIED-DATE = July  10, 1980;
SOCIAL-INSUR-NO = 888 888 888;
BIRTHDATE = Sept. 14, 1945;
BIRTHPLACE = Hoboken, New York;
MARITAL-STATUS = Married;
NUMBER-OF-DEP = 1;
F-PRE-NAME = Aardvark, Agnes, Mrs.;
F-PRE-NAME = Aardvark, Anthony, Mr.;
  RELATIONSHIP = son;
  F-BIRTHDAY = June  16, 1979;
ORG-NAME = E.P.S.B.;
  W-ADDR-TYPE = 3;
    W-STREET = 10010-107A Ave.;
    W-CITY = Edmonton;
    W-PROVINCE = Alberta;
    W-COUNTRY = Canada;
    W-POSTAL-CODE = T4G 0N0;
    W-PHONE = 429-5621;
    W-VERIFIED-DATE = July  10, 1980;
  JOB-TITLE = Teacher;
    JOB-DATE = 1974 to present;
    JOB-DES = Teaching ;
    EDUC-SPEC = Teaching;
      STUDENT-TYPE = Grade 1-3;
ISSUE-AUTHORITY = ASTD;
  TYPE-OF-CERT = 999999;
  T-DURATION = Permanent;
  DATE-OF-ISSUE = Sept. 1, 1978;
DEGREE-OBTAINED = B.Sc.;
  INSTITUTION = University of Alberta;
  SPECIALIZATION = Education;
  YEAR-COMPLETED = 1967;
ASSISTANCE-START = Sept.  1, 1980;
  ASSISTANCE-END = April 30, 1981;
GRANTING-AGENCY = Queen Elizabeth Scholarship;
  AWARD-TYPE = Scholarship;
  AWARD-AMOUNT = $500.00;
  AWARD-PERIOD = 1967;
```

```
  AWARD-STATUS = Completed;
  AWARD-OBTAINED = University of Alberta;
PUBLICATION-DATE = July  1979;
  TITLE = The Engineer as a Teacher;
  CO-AUTHOR = Smith, W.P.;
  PUBLISHER = The Education Digested;
  PAGES = 23-43;
DEGREE = MASTERS;
  PROGRAM = Thesis;
  CURRENT-STATUS = Candidate for Masters;
    DATE-STATUS-MOD = July  10, 1980;
  ENROLL-STATUS = Full Time;
    DATE-ENROL-MOD = July  10, 1980;
  PRINT-COMMENTS = YES;
    COMMENTS = This is a dummy student ;
  ADVISOR-ID = 24;
    ADVISOR = Seger;
    DATE-APPOINTED = July  10, 1980;
UA-COURSE = EDPSY502;
  UA-COURSE-COM = This course has been granted equivalent
                  to EDADM511/512 ;
  YEAR = 1976;
    SESSION = Spring;
      SECTION = A3 ;
        UA-MARK-TYPE = MA;
          UA-MARK-DATE = July  10, 1980;
          UA-MARK = 9;
        CREDIT-PROG = MASTERS;
        COURSE.STATUS = Advanced Credit;
        UA.TRAN.CAT = Special Student;
          UA.TRAN.FAC = Education;
```

Figure 4

MODIFYING A STUDENT RECORD

```
  . . .
  . . .
  . . .
:Please enter the number of an option: 1
    *
    *        DEPARTMENT OF EDUCATIONAL ADMINISTRATION
    *               COMPUTER SUPPORT SYSTEM
    *
    *  Individual Record Addition and Modification Section
    *  ********** ****** ******** *** ************ *******
    *
:Enter Student Surname or ID: 631791
    *
    *
    *
    *
    *
    * The following options for modifying the student record are
    * available:
    *
    *  1. Change the ID number for the student.
    *  2. Add a new address for the student.
    *  3. Enter the information on the Departmental Application
    *     Form.
    *  4. Change information on the Current Registration of the
    *     Student.
    *  5. Enter all the Personal Information (Name, Address,
    *     etc.) for a student whose file exists already (usually
    *     because we have marks already on file for this student)
    *  6. Enter Advanced Credit information for this student.
    * 98. Output a formatted copy of student file (DEA/S/01).
    * 99. Edit the student's file.
:Enter the number of an option (ATTN if Finished):
  . . .
  . . .
  . . .
```

each record and by which the record itself will be stored.

If a student ID is incorrect, then the complete record must be removed from the data base and re-entered with the correct ID. This option does this for the user, without the user being aware of all that is involved. Interactive DEACSS merely prompts for the new ID, then does all the work.

2.2.2.2 Add a new address for the student

Since our students are moving, graduating, etc. this option allows us to add a new address for the student. The new address will be coded as to a "University", "Work", or "Permanent" address. Previous addresses are currently kept in the file, but addresses are stored so that the most recent address is the first address found in the file.

2.2.2.3 Enter the information on the Departmental Application Form

This option exists in case a student file has been built for a student before the Departmental Application Form has been received. Interactive DEACSS prompts for the information from this form exactly as was done for the Departmental Application for Admission form for Aardvark in Figure 2.

2.2.2.4 Change information on the Current Registration of a Student

The Current Registration of a student is the Degree or Diploma Program which that student is currently pursuing or was

pursuing during their last valid registration in the Department. Currently we have four categories of students in the Student File:

1. Ph. D.
2. M. Ed.
3. Diploma
4. Others not registered in the Department

Only the first three students actually have a valid Current Registration in the DEACSS. The options available for changing the Current Registration of the student are shown in Figure 5. Each option in described below.

1. **Enter a completely new registration**

   This option is used when we have an existing record for a student but that student is enrolling in a different degree. The most obvious example is when a student who has obtained a M. Ed. enrolls in the Ph. D. program. If the student has not yet convocated, the system will first ask you to complete the information for the degree in which the student is currently enrolled.

   The system will then prompt you for the information unique to the program in which the person is now registering.

2. **Change the Current Status within this registration**

   The Current Status of a student is the type of registration the student currently holds, (eg. Candidate for Ph.D., Qualifying Graduate Student, etc.) Depending on the Degree for which the student is registered, Interactive DEACSS will give you the valid options for Current Status,

Figure 5

MODIFYING A CURRENT REGISTRATION

```
 . . .
 . . .
 . . .
   *            CURRENT REGISTRATION OPTIONS
   *            ******* ************ *******
   *
   *  1. Enter a completely new registration (eg. change from
   *     Masters to Ph. D.).
   *  2. Change the current status within this registration
   *     (eg. change from "Provisional Candidate" to
   *     "Candidate").
   *  3. Change the Enrollment status within this registration
   *     (eg. change from "Full Time" to "Part Time").
   *  4. Initiate or change the student's Advisor.
   *  5. Add or change information on a Committee Member.
   *  6. Enter or change the status of a Comment on this
   *     registration.
   *
   *
   *            *** ATTN to return ***
   *
:Please enter the number of an option
 . . .
 . . .
 . . .
```

then ask you to choose one. You will also be prompted for an official date for the change in status.

If the new status of the student is "Convocated", you will be asked for the month of convocation, title and keywords for the thesis (if a thesis was written). When the Current Status of a student is changed to "Convocated", the Enrollment Status is also changed automatically.

3. **Change the Enrollment Status within this registration**

The Enrollment Status for a student is usually Full Time, Part Time, Withdrawn, or Convocated (although others may exist). Interactive DEACSS will give you the existing Enrollment Status for the student and the valid options for Enrollment Status, then ask you to choose one. You will also be prompted for an official date for the change in status.

4. **Initiate or change the student's Advisor**

This option tells you who is currently assigned as the student's advisor (if an advisor is assigned), then asks you for the new advisor and the date of the appointment. If an advisor existed already the date of the appointment of the new advisor is the date used for the termination of the old advisor.

5. **Add or change information on a Committee Member**

This option allows you to add, change the status of, or delete a member of a student's committee. Once again you will be prompted for a date for the action.

6. **Enter or change the status of a Comment on this registration**

Sometimes a student will be registered in a particular

degree, but a caveat will be placed upon the registration (eg. subject to the student completing Ed. Adm. 511-512 by ...). This option allows us to enter such comments, or to indicate that the requirement has been successfully met. As with all information in a student file, a comment will never be withdrawn from the file, but an indication will be given if it is still in force.

2.2.2.5 Enter all the Personal Information for a Student File which already exists

In some cases, we will have a student enroll in the Department who has already taken a course from the Department. In this case, we will have none of the personal information in DEACSS, only the student marks.

This option is very dangerous. If we enter all new information for a student, then find out that we had the wrong student ID, we will have to enter the information on the first student again! This option warns you that this can be a potentially dangerous move. It is suggested that before using this option, use option 99 to Edit the student file simply to look at what is in the file so that you can be sure you are not about to overwrite a perfectly good file.

If you choose this option, a similar session to that shown in Figure 2 will occur.

2.2.2.6 Enter Advanced Credit Information

Advanced Credit for courses can be added through this option. This option follows the Department of Graduate Studies

Recommendation for Advanced Credit form. Courses can be entered which were taken either at the University of Alberta, or at other institutions.

Exactly the same procedure is used as was used in Figure 2 to prompt for Alan Aardvark's advanced credit in EDPSY502.

2.2.2.7 Output a formatted copy of the Student File (DEA/S/01)

Once any modifications have been made to the Student File, you may wish to generate a clean paper copy of all the information in the file. This could be of use to the student's advisor, Department Chairman, etc. This option will generate a copy of the student file formatted nicely (Format DEA/S/01). An example of the formatted copy of the file for Mr. Aardvark is shown in Appendix E.

2.2.2.8 Use the MTS Editor to modify a student record

The use of the MTS Editor to modify a student record is the option of last resort. It should be used only if no other option in Interactive DEACSS exists to modify the student record, or if the modification is trivial (eg. spelling corrections, etc.).

If you must use the MTS Editor, due to the amount of data in a student record, and the possibility of error, it is recommended that the visual editor be used.

1. Once Interactive DEACSS has placed you in the Editor, type "v".
2. In the visual mode, "what you see is what you get".
3. Leave the visual mode by hitting the "ATTN" or "PA1" key.

4. Type "stop" to exit from the editor and to return control to Interacive DEACSS.
5. You will be asked if you wish to update the Student Record. If you made changes and wish them to be made permanently stored in the student file, respond "yes". If you did not make any changes, or if you had a disaster occur while editing, respond "no" and the permanent record will not be changed.
6. If you responded that you did want the file permanently updated, Interactive DEACSS will attempt to perform this update. If your changes were *syntactically* correct, they update will be made, and you will receive the message:

   Congratulations!! File Updated Correctly

   and given the option of working on another student record or returning to the Main Menu.

   If the update was not successful, you will be told what syntactic errors were detected, and at which line in the file these occurred. You will be asked if you want to attempt to correct the errors, or if you would like to abort the change. If you select the option to attempt to correct the errors, you will be placed in the MTS Editor once again, and the process will be repeated. If errors occurred which you can not understand, select the option to abort the modification and contact the author or a SPIRES consultant.

CAUTION - When using the Editor:

1. **NEVER** change the Student ID number.
2. Do not modify the Data element identifier (to the left of the

"=" sign.)

3. Each Data element MUST be terminated by a ";".
4. If you are adding new data elemtents or structures, each element or structure must be placed in the correct position, and each element identifier must be spelled correctly. Section 5.3 explains the adding of new structures and elements to DEACSS.

2.2.3 Completion of a Student Record Modification or Addition

After you have added or modified each record, the program will give the option of either:

1. Entering or modifiying another student record.
2. Returning to the Master Menu for Interactive DEACSS.

2.3 Adding A Staff Record to DEACSS

New Staff members can be added to DEACSS in two ways:

1. Through the master menu, or
2. By attempting to assign someone as an advisor or a committee member, when that person is not yet in the staff file.

In both these cases, Interactive DEACSS goes to the same routine to prompt for a *minimal* amount of information on the staff member. Figure 6 shows an example of the interaction which occurs when the staff member is in the Department of Educational Administration. The simple record in the STAFF FILE which is created by this interaction is shown in Figure 7. There is certain other information, such as home address, home phone number and research interests which can be kept in the Staff Record. These should be added using Native SPIRES at some convenient time. Section 5.4 shows how to expand the simple

Figure 6

ADDING A STAFF RECORD

```
*
*
*
*                  DEPARTMENT OF EDUCATIONAL ADMINISTRATION
*                          COMPUTER SUPPORT SYSTEM
*
*   MAIN MENU
*   **** *****
*
*   Currently, you have the following options:
*    1) Add or modify individual student records.
*    2) Add a new staff record.
*    3) Enter a group of students requesting
*       assistanceships.
*    4) Generate a Report.
*    5) Generate *TEXTFORM compatible output.
*   98) Enter Native Spires Mode (do not processthe
*       file).
*   99) Return to MTS (do not process the file).
*
*       To terminate the run normally, hit "ATTN" or "PA1".
*

:Please enter the number of an option: 2
    * I am creating a new record for this staff member.
    *
:Please enter the name (Last, First Second, Title)
                          Montgomerie, Thomas Craig, Mr.
:Is this person at the University of Alberta? yes
    *
    * Academic Ranks are:
    * 1 - Professor
    * 2 - Associate Professor
    * 3 - Research Associate Professor
    * 4 - Assistant Professor
    * 5 - Research Assistant Professor
    * 6 - Sessional Lecturer
    * 7 - Sessional Lecturer (part time)
    * 8 - Graduate Assistant
    * 9 - Other
:What is the rank of this staff member 7
:Is this person in the Faculty of Education? yes
    *
    * In which department is this person?
    * 1 - Department of Educational Administration
    * 2 - Department of Educational Psychology
    * 3 - Department of Elementary Education
    * 4 - Department of Secondary Education
    * 5 - Department of Industrial and Vocational Education
    * 6 - Department of Educational Foundations
```

```
    * 7 - Unknown
:Please enter the number of the Department 1
:Is this person currently active in the Department? yes
:Please enter their Office Number: 3-104 Ed. N.
:Phone? 432-3762
    *
    *   Currently, you have the following options:
   . . .
   . . .
   . . .
```

Figure 7

A SAMPLE STAFF RECORD AFTER INITIAL ADDITION

```
STAFF-ID = 27;
STAFF-NAME = Montgomerie, Thomas Craig, Mr.;
STAFF-ACTIVE = YES;
STAFF-RANK = Sessional Lecturer (part time);
DEPT = Department of Educational Administration;
STAFF-OFFICE = 3-104;
STAFF-PHONE = 432-3762;
```

record in Figure 7 to a complete staff record.

2.4 Students Requesting Assistanceships

Interactive DEACSS usually works on one student or Staff record at a time. There are occassions when we require that exactly the same change be made to a number of records. One example of this occurs when students are asked if they require assistanceships for the next academic session. When this occurs, the department receives a number of requests for assistanceships for exactly the same period.

In order to facilitate entry of this data, Interactive DEACSS first queries for which dates the assistanceship is requested, then asks for a list of student names or ID numbers making that request. Interactive DEACSS checks each student name or ID for validity, and if it is valid, updates each record in turn. This method of updating results in significantly reduced time and keystrokes.

2.5 Generating Reports on Interactive DEACSS

Certain reports are required repeatedly in the Department. These reports may be as simple as lists of students currently enrolled in the Masters Program, or as complex as Departmental Workload Reports. The kind of data to be included in these reports are the same each time the report is required, but the membership of the groups may change. In order to facilitate the generation of these reports, Option 2 in the MASTER MENU of Interactive DEACSS provides for the generation of these reports.

Figure 8 gives the Report Menu for Interactive DEACSS. When the particular report required is selected, Interactive DEACSS will prompt for the parameters which describe the group to be included in that report, the number of copies of the report to be generated and whether the report should be printed on the Xerox 9700 page printer located in the Department of Computing Services or on the remote line printer located in Room 3-121 Education North.

Examples of all the reports are given in Appendix E. Each report has a number in the top right corner of the page. This is a number unique to that report type, and is the easiest way for someone requesting a report to specity which report format they wish.

The information requested by Interactive DEACSS in order to produce each report will vary slightly, but the information should all be obvious to the person requesting the report. Figure 9 gives a sample interaction which produces report DEA/S/04 - a list of students, their addresses and their advisor - for full-time M.Ed. students.

2.5.1 A Note on Security

The remote line printer in Room 3-102C Education North is located in a publically accessible area, and prints at a speed at which information could be read as it is produced. Material produced by the XEROX 9700 page printer is returned to Room 3-102C usually within one-half working day of the time it is requested. The material is placed in a brown envelope and is

Figure 8

REPORT OPTIONS

```
    *
    *
    *
    *                DEPARTMENT OF EDUCATIONAL ADMINISTRATION
    *                        COMPUTER SUPPORT SYSTEM
    *
    *   REPORT GENERATION MENU
    *   ****** ********** ****
    *
    * The reports which can be generated are:
    *    1) DEA/S/01 - Copies of the complete student file
    *                  for a group of students .
    *    2) DEA/S/02 - A list of student addresses.
    *    3) DEA/S/03 - A list of students with advisors.
    *    4) DEA/S/04 - A list of student addresses and advisors.
    *    5) DEA/S/05 - A list of students with no advisor.
    *    6) DEA/S/06 - A list of all students requesting
    *                  assistance.
    *   51) DEA/F/01 - A staff list with rank, office and phone.
    *   52) DEA/F/02 - A list of staff with their advisees, and
    *                  students on whose committees they serve.
    *   53) DEA/F/03 - A staff list with home address and phone.
    *   54) DEA/F/04 - A staff list with interest, office
    *                  and phone.
    *   55) DEA/F/05 - A Departmental Workload Report.
    *
    *      To return to the MAIN MENU press "ATTN" or "PA1".
    *
:Please enter the number of an option:
```

Figure 9

GENERATION OF A REPORT

```
   *
   *
   *
   *                 DEPARTMENT OF EDUCATIONAL ADMINISTRATION
   *                         COMPUTER SUPPORT SYSTEM
   *
   *   REPORT GENERATION MENU
   *   ****** ********** ****
   *
   * The reports which can be generated are:
   *   1) DEA/S/01 - Copies of the complete student file
   *                 for a group of students .
   *   2) DEA/S/02 - A list of student addresses.
   *   3) DEA/S/03 - A list of students with advisors.
   *   4) DEA/S/04 - A list of student addresses and advisors.
   *   5) DEA/S/05 - A list of students with no advisor.
   *   6) DEA/S/06 - A list of all students requesting
   *                 assistance.
   *  51) DEA/F/01 - A staff list with rank, office and phone.
   *  52) DEA/F/02 - A list of staff with their advisees, and
   *                 students on whose committees they serve.
   *  53) DEA/F/03 - A staff list with home address and phone.
   *  54) DEA/F/04 - A staff list with interest, office
   *                 and phone.
   *  55) DEA/F/05 - A Departmental Workload Report.
   *
   *      To return to the MAIN MENU press "ATTN" or "PA1".
   *
:Please enter the number of an option: 4
 * You can produce lists for any of the following students:
 * 1      FULL TIME PH.D.
 * 2      PART TIME PH.D.
 * 3      ALL PH.D.
 * 4      FULL TIME MASTERS
 * 5      PART TIME MASTERS
 * 6      ALL MASTERS
 * 7      THESIS MASTERS
 * 8      NON THESIS MASTERS
 * 9      ADMINISTRATIVE DEVELOPMENT PROGRAM
 * 10      TEACHING SKILLS IMPROVEMENT PROGRAM (FULL TIME)
 * 11      FULL TIME MASTERS (NO ADP OR TSIP)
 * 12      PART TIME MASTERS (NO ADP OR TSIP)
 * 13      ALL MASTERS (NO ADP OR TSIP)
 * 14      ALL ACTIVE STUDENTS
 *
:Please enter the number of your choice: 4
 *
 * You can have this output:
 * 1. Typed on the terminal
 * 2. Printed downstairs
```

```
 * 3. Printed on the Page Printer at the Computer Center
 *
:Please enter your option: 3
:How many copies would you like? 1
# *PRINT* ASSIGNED RECEIPT NUMBER 839019
# *PRINT* 839019 HELD
# *PRINT* 839019 RELEASED, 10 PAGES
   .   .   .
   .   .   .
   .   .   .
```

secure from casual information seekers. It is strongly recommended that, unless a report is required immediately (within 4 hours), all reports be directed to the XEROX 9700 page printer. This provides some measure of security on this printed information as well as providing much nicer printed output.

2.6 Generating *TEXTFORM Compatible Output

Currently there is only one Spires Format which generates *TEXTFORM compatible output. This Format is called TEXT-LETTER, and has been used in generating individual letters and questionnaires to students. The way we interface to *TEXTFORM is to first generate a *TEXTFORM macro "&NAME" which takes a number of parameters. These parameters are the individualized portion of the letter or questionnaire (eg. the student name or address).

The Spires Format TEXT-LETTER calls the *TEXTFORM macro "&NAME" and passes 13 parameters to it. An example of macro &NAME and its parameters are shown in Table 1. Examples of the Spires output from TEXT-LETTER, the use of this macro and examples of the individualized output are shown in Appendix F.

When you choose the option to generate *TEXTFORM compatible output, the only difference from generating a report will be that instead of asking you upon which printer you wish the output generated, you will be asked to give the name of a MTS file in which to place the output.

TABLE 1

PARAMETERS OUTPUT BY SPIRES FORMAT TEXT-FORMAT

USED BY SPIRES MACRO &NAME

Parameter	Description	Example
1	Honorific	Mr.
2	Initials	T.C.
3	Surname	Montgomerie
4	Street Address	11647 - 77 Ave.
5	City, Province and Country	Edmonton, Alberta <NL>Canada
6	Postal Code	T6G 0M4
7	Preferred Name	Craig
8	Advisor's Name	Dr. J.E. Seger
9	Phone	Phone: 436-2628
10	Social Insurance Number	606 268 894
11	U of A ID	631791
12	Degree	M.Ed.
13	M.Ed. Program	Thesis (blank if Ph. D.)

EXAMPLE OF &NAME USING PARAMETERS GIVEN ABOVE

```
<&NAME(' Mr.',' T.C.','Montgomerie',-
'3-104 Education North, U of A',-
'Edmonton, Alberta<NL>Canada','T6G 2G5',-
'Craig','Dr. J.E. Seger',-
'Phone: 432-3762','606 268 894','631791','M.Ed.','Thesis')>
```

2.7 Exiting Interactive DEACSS

There are three ways to exit from Interactive DEACSS:

1. Striking the Attention Key when you are asked to select an option in the Main Menu. This is the normal way you should exit from Interactive DEACSS.

   When this method of exiting Interactive DEACSS is chosen, the program will automatically "Process" the file (ie. perform the operations necessary to change the file to permanently incorporate the changes or additions made during the session.) As this procedure is performed, a few lines of information are printed which indicate how many records are being added or updated.

   After Interactive DEACSS has completed this operation, you will be returned to MTS. MTS will print the "#" prompt to indicate this. You may then signoff the system by typing:

   $SIGNOFF

2. Choosing Main Menu Option 98 places you immediately into Native Spires. At this point you can issue standard Spires commands.

   If this method is chosen, any changes or additions to the file will not have been "Processed".

3. Choosing Main Menu Option 99 immediately transfers you back to MTS without "processing" the file. This option should only be chosen when you know that nothing you have done in any way changes or adds to DEACSS.

NOTE: The first option, that is hitting the Attention key is by

far the preferred method of exiting Interactive DEACSS.

2.8 Modifying a Student Record Immediately After Entry

If you wish to modify a Student file during the same session in which it was entered, you *MUST* either,

1. Respond with the Student ID *only* when Interactive DEACSS prompts for the Surname or the Student ID, or
2. terminate the session normally (using the Attention), then issue the command:

   ```
   $SOURCE CONTROL
   ```

   once more.

It is not possible to reference a new record by any means other than the Student ID until it has been permanently added to the file (which is done when you respond that you have no more students to add or modify).

3. Adding Courses and Student Marks to DEACSS

Student marks are usually added to DEACSS in large groups. In order to allow for verification of entry, and to decrease costs, it has been decided to keypunch Department of Educational Administration mark information. This data is then entered to DEACSS via a Batch (off-line) procedure.

3.1 Keypunching the Mark sheets

The Course mark (or Course Registration) class lists will be keypunched. Two types of cards will be punched for each Class List. The format for these two types of cards are shown in Tables 2 and 3.

a. Table 2 - The Course Information Card (one card per course).

b. Table 3 - The Student Mark Card (one card for each student in course).

TABLE 2

FORMAT FOR COURSE INFORMATION CARD

Card Column	Description	Example
1- 8	Course Name and Number	EDADM461
10-15	Session in which the course begins (ie. FALL, WINTER, SPRING, SUMMER)	WINTER
17-20	Year in which course begins	1975
22	Term in which course is offered	1
24-26	Section	A1
27-30	ID of Teacher offering course	20
32-40	Time course offered (24 Hour clock)	1230-1400
41-45	Days on which course is offered (ie. M, T, W, R, F, S, MWF, MW, TR, MTWRF)	TR
46-63	If on campus - Room If off campus - City	1-128

TABLE 3

FORMAT FOR STUDENT MARK CARD

Card Column	Description	Example
1- 8	Course Name and Number	EDADM461
10-15	Session in which the course begins (ie. FALL, WINTER, SPRING, SUMMER)	WINTER
17-20	Year in which course begins	1975
22	Term in which course is offered	1
24-26	Section	A1
27-32	Student Id	631791
34-72	Student Name	HALL, S.
73-74	Mark Field	2
76-77	Recommendation Field	F
79-80	RE - if Re-mark EN - if Enrollment/Registration MA - if Mark	MA

Before sending the Course Mark or Course Registration class lists to the keypuncher, some information not normally found on these lists must be added. This information, which is available on the Course Enrollment Forms, or from the Registration Handbook, follows:

1. The Teacher ID - write the ID assigned to the instructor beside the Teacher's name. If the instructor has not been assigned an Teacher ID, you must add the instructor to the data base *before* you attempt to add a course having that instructor.
2. Session in which the course begins. You must mark on the front of the class list the session (FALL, WINTER, SPRING, SUMMER) in which the course begins.
3. Term in which the course is offered. This will be either 1, 2 or F (for full term). Assure that class lists are correctly noted.
4. Time course is offered. Write the time of offering (using the 24 hour clock) on the top of the class list.
5. Days on which course is offered. Days of the week for which the course is offered are:
   a. M for Mondays only
   b. T for Tuesdays only
   c. W for Wednesdays only
   d. R for Thursdays only
   e. F for Fridays only
   f. S for Saturdays only
   g. MWF for Monday, Wednesday, Friday, etc.

6. Location of course:

   a. Room number if on Campus

   b. City of town if off campus

A rubber stamp will be prepared for stamping the class lists to add the data not automatically included.

3.2 Adding courses and marks to the Data Base

After the cards have been keypunched, the following job must be run in BATCH mode to add this information to the Data Base.

```
 1) $SIGNON SPIS 9TP=1 PRIO=L RETURN=EDUC T=1M P=300
 2) Password
 3) $EMPTY MARKFILE
 4) $EDIT MARKFILE
 5) INSERT
 6) Put data cards here
 7) $ENDFILE
 8) STOP
 9) $MOUNT E00000 9TP *T* VOL=MARKS1 RING=IN
10) $RUN *FS 0=*T*
11) SAVE MARKFILE MARKS.session.year
12) STOP
13) $SOURCE SPIS:ADD.MARKS
14) january 15, 1975
15) $SIGNOFF
```

A deck which performs this job has been provided to the Department office. You should modify the deck in the following way:

1. Insert the new data cards in 6).
2. Change the card in 11) to be the current session and year being added or updated.
3. Change the card in 14) to be the current date, or in the case where you are adding marks from previous years, the date on which the marks were official.

4. Interfacing DEACSS with TEXTFORM

TEXTFORM is a Text formatting program written and supported by the Department of Computing Services at the University of Alberta. One of the nice features of TEXTFORM is the ability to generate "personalized form letters". These are similar to the form letters received from many direct mailing distributors, who print thousands of letters with selected blank spaces, then type some kind of personalized message in the blank space to make each recipient feel the letter was written to them personally. Usually the type face does not match or the ribbon is not as dark as the printed copy.

In TEXTFORM we can use the MACRO facility to generate the same kind of letter, but each letter is produced individually with the personalized information fitted into the letter before each letter is printed.

We can have letters printed on Departmental letterhead, colored stock or plain paper on either the Anderson Jacobson 832 or on the Xerox Page Printer. Personalized envelopes can be produced on the Anderson Jacobson 832.

There is currently one FORMAT built in DEACSS which is designed specifically to provide an interface to TEXTFORM. This format is called DEA/S/12 and is called as Option 15 of the Main Menu in Interactive DEACSS. This format can also be specified from native SPIRES. This Format generates a call to a TEXTFORM MACRO called "&NAMES". &NAMES has 13 possible parameters, which are defined in Table 1.

Each student who has been selected from DEACSS will generate an individual call to the macro &NAMES. The user should write the TEXTFORM macro &NAMES which will generate the appropriate letter using the parameters passed to it by the calling procedure. The last line in the file in which the TEXTFORM macro &NAMES is defined should be a

$CONTINUE WITH file

command, where "file" is the name of the file where you told Interactive DEACSS to place the TEXTFORM macro calls.

Appendix F contains:

1. an example of the output of FORMAT DEA/S/12 (the TEXTFORM compatible output format) for two students.
2. three examples of possible TEXTFORM macro &NAMES which produce:
3. a individualized letter which informs students of the agenda for the 1980 Graduate Student Registration meetings, and asks them to fill in an attached questionnaire on their 1980-81 registration.
   a. a form which asks students to provide some information on their plans for the coming academic year and to verify our records, and
   b. an envelope.
4. The MTS jobs which produced the actual letters, questionnaires and envelopes.
5. Samples of the output from those three TEXTFORM runs.

The TEXTFORM commands for the macro &NAMES which produces the letter above is in the file SPIS:LETTER. This letter was printed

on Departmental letterhead using the XEROX printer at the Department of Computing Services and was mailed to each currently enrolled student

The TEXTFORM commands for the macro &NAMES which produces the questionnaire is in the file SPIS:QUEST. The questionnaire was printed on blue stock using the XEROX printer at the Department of Computing Services and was attached to the letter.

The TEXTFORM commands for the macro &NAMES which produces the envelopes is in the file SPIS:ENVELOPE. Envelopes were printed on the Anderson Jacobson 832 terminal.

5. Accessing DEACSS From Native Spires

As the members of the Department use DEACSS, their requirements constantly evolve. Interactive DEACSS has been designed to provide the features which are currently requested by members of the Department, but requests for information presented in different forms, or different combinations of information are constantly expected and must be accommodated.

Before attempting to use DEACSS from Native SPIRES, the user should have the references given in the "Technical References" Section, and should have perused the following:

Culham, E. *The File Editor*. Edmonton: Department of Computing Services, The University of Alberta, 1979.

Jackson, G.R. *SPIRES Searching and Updating*. Edmonton: Department of Computing Services, The University of Alberta, 1978.

Senda, R.E. *SPIRES/370 Data Base Management*. Edmonton: Department of Computing Services, The University of Alberta, 1978.

It is also recommended that the user take the courses entitled

"Introduction to SPIRES", and

"SPIRES Data Base Management"

which are offered by the Department of Computing Services at the University of Alberta.

5.1 Selecting the Correct SPIRES Subfile

The procedure for accessing DEACSS is:

1. Sign on to the Computer:

```
$SIGNON SPIS
```

2. Enter the Password:

    The password will be made available to a valid user by the author.

3. Enter SPIRES by means of the command:

```
$RUN *SPIRES
```

4. Currently DEACSS contains three SPIRES subfiles:
    a. a student file (GRAD RECORDS)
    b. a staff file (STAFF FILE)
    c. a course file (COURSE FILE)

    You should select the subfile for the kind of information you wish. For instance, to access the student file, you would issue the command:

```
SELECT GRAD RECORDS
```

5. You can now find out all the searchable indices by typing the command:

```
SHOW INDICES
```

    or you can get a list of all the data elements and structures in the file by typing the command:

```
SHOW ELEMENT DICT
```

5.2 Using the FIND Command

If the data you want is indexed (ie. the element shows up when you issue the "SHOW INDICES" command), you can issue a "FIND" command to get the appropriate data. For example, if you wanted to find all the students by the name of "Montgomerie" who are in the file, you would issue the following command:

FIND NAME MONTGOMERIE

DEACSS would respond with

-Result: 2 GRAD-STUDENT(S)

You then have numerous options:

6. You could view the records for these students on the terminal by issuing the command:

   Type

7. You could put these records into the SPIRES Active File by using the command:

   Output

8. You could use the "DEFINE TABLE" command to get a very rudimentary table output of these students. For example, if you wanted to find out the students names and student ID numbers, you would issue the following commands:

   FIND NAME MONTGOMERIE

   DEFINE TABLE IDNO(1,10) NAME

   TYPE

   which would result in the following output:

```
July 19, 1980                                        PAGE 1
  IDNO        NAME

  640744      Montgomerie, Heather Lynn, Mrs.
  631791      Montgomerie, Thomas Craig, Mr.
```

If you were to issue the commands:

FIND NAME MONTGOMERIE

```
DEFINE TABLE IDNO(1,10) NAME
OUTPUT
```

exactly the same table would be output to the Spires Active File.

5.2.1 Special Indices in the Subfile GRAD RECORDS

An Index in SPIRES is usually defined so that each time an element in a SPIRES record takes a value, a pointer to that record is generated in the Index. Occasionally we do not wish to have all values in a particular element placed in the Index, but only the most recent or current value for that element.

Our GRAD RECORDS file is a historical file of all students. Since we maintain all information ever entered for a student, a particular student who, for instance, graduated from the Ph.D. program might have had all the following entered in his file as his current status:

1. Applied for admission to Ph.D.
2. Provisional Candidate for Ph.D.
3. Candidate for Ph.D.
4. Convocated

If we had generated a normal Index to Current Status, we would not be able to easily find only those people who were, for instance, currently registered as Candidates for Ph.D.

In order to eliminate this problem, the indices:

1. ENROLLEMENT STATUS
2. CURRENT STATUS
3. DEGREE

have been designed so that the *latest* entry for each (the first

entry shown when the record is displayed) is entered in the Index. Thus, our Ph.D. student would only be found if we asked for those students *who had convocated*.

5.3 Adding Structures and Elements to DEACSS

Appendix D shows the Summary Data Element Dictionary for each of the three subfiles in DEACSS (GRAD RECORDS, STAFF FILE and COURSE FILE). In order to add a data structure (or element to a structure), to a record, the first step is to find out in which SPIRES subfile the data should reside. You will SELECT this subfile.

Once you have SELECTED the appropriate subfile in which to put the data, you must then find the Record ID for the record to which you wish to add the data. In the GRAD RECORDS file the Record ID is the Student's U of A ID number, in the Staff Record, it is a Staff Record number (see section 5.3), while in the COURSE FILE it is the Official U of A course number (eg. EDADM511). You will TRANSFER the appropriate record into the Spires Active File.

At this point the record can be modified. The MTS Editor should be used to insert new data structures and data elements. **Data structures and elements must be placed in the correct place in the record or an error may occur. Always refer to Appendix D for the correct placement of Data Elements or Structures in a Record.** In DEACSS we have set the custom that the newest data should be placed *before* existing data. We have also set a convention that *no data will be removed from a record unless it*

was incorrect.

Once the data has been added to the record we must UPDATE the file. SPIRES will inform you if there are any syntactic errors, but semantic errors are your problem. If there are errors, you can EDIT the Spires Active File again to these errors and UPDATE again.

Note: You *must* spell the name of each data element correctly, the data element name must be followed by an "=" sign, and the data element value must be terminated with a ";". A ";" must never appear in a data element other than to terminate that value. See the SPIRES REFERENCE MANUAL if you wish to put quotes ("), apostrophes (') or other special symbols in a data element.

Figure 10 gives an example using this method to add a Citizenship structure to a student record (good old Alan Aardvark!!).

5.4 Modifying A Staff Record

Staff records are usually changed on an infrequent basis. Currently the information in a staff record can only be modified from Native SPIRES. In order to modify any record to Native SPIRES, you must know the key to that record. In the STAFF file, the record key is the STAFF-ID. There are two ways to find out the key to a staff file.

1. Produce a list of all staff members and their STAFF-ID .
2. "FIND" the individual record and "TYPE" the record to see the key

The first of these is recommended if you will be making a number

Figure 10

SAMPLE INTERACTION TO MODIFY A STUDENT RECORD

```
# r *spires
# 20:54:16
 Spires active file set to: -SAF
 -Spires 6.0
 -Tracing on, 'EXPLAIN TRACING' for details
-? select grad records
-? trans 808080 clear
-? edit
   Editing file -SAF
: v
```

(AT THIS POINT THE FOLLOWING WOULD APPEAR ON THE SCREEN)

```
   1      IDNO = 808080;
   2      LEGAL-NAME = Aardvark, Alan A., Mr.;
   3      PREFERRED-NAME = Alan;
   4      MODDATE = July  10, 1980;
   5      DEPARTMENT = Educational Administration;
   6      S-ADDR-TYPE = 1;
   7        S-STREET = 999 Llama Road;
   8        S-CITY = Edmonton;
   9        S-PROVINCE = Alberta;
  10        S-COUNTRY = Canada;
  11        S-POSTAL-CODE = T6G 0V0;
  12        S-PHONE = 499-9999;
  13        S-VERIFIED-DATE = July  10, 1980;
  14      SOCIAL-INSUR-NO = 888 888 888;
  15      BIRTHDATE = Sept. 14, 1945;
  16      BIRTHPLACE = Hoboken, New York;
  17      MARITAL-STATUS = Married;
  18      NUMBER-OF-DEP = 1;
  19      F-PRE-NAME = Aardvark, Agnes, Mrs.;
  20      F-PRE-NAME = Aardvark, Anthony, Mr.;
  21        RELATIONSHIP = son;
  22        F-BIRTHDAY = June  16, 1979;
  23      ORG-NAME = E.P.S.B.;
  24        W-ADDR-TYPE = 3;
  25          W-STREET = 10010-107A Ave.;
  ..            . . .
  ..            . . .
  ..            . . .
```

(YOU WOULD MODIFY THE RECORD USING THE VISUAL EDITOR SO THAT THE FOLLOWING APPEARED ON THE SCREEN)

```
   1      IDNO = 808080;
   2      LEGAL-NAME = Aardvark, Alan A., Mr.;
   3      PREFERRED-NAME = Alan;
   4      MODDATE = July  10, 1980;
   5      DEPARTMENT = Educational Administration;
   6      S-ADDR-TYPE = 1;
   7        S-STREET = 999 Llama Road;
   8        S-CITY = Edmonton;
   9        S-PROVINCE = Alberta;
  10        S-COUNTRY = Canada;
  11        S-POSTAL-CODE = T6G 0V0;
  12        S-PHONE = 499-9999;
  13        S-VERIFIED-DATE = July  10, 1980;
  14      SOCIAL-INSUR-NO = 888 888 888;
  15      BIRTHDATE = Sept. 14, 1945;
  16      BIRTHPLACE = Hoboken, New York;
  17      MARITAL-STATUS = Married;
  18      NUMBER-OF-DEP = 1;
  19      F-PRE-NAME = Aardvark, Agnes, Mrs.;
  20      F-PRE-NAME = Aardvark, Anthony, Mr.;
  21        RELATIONSHIP = son;
  22        F-BIRTHDAY = June  16, 1979;
  22.1    CITIZEN-STR;
  22.2    COUNTRY = Canada;
  22.3    C-DATE = July 1, 1980;
  23      ORG-NAME = E.P.S.B.;
  24        W-ADDR-TYPE = 3;
  25          W-STREET = 10010-107A Ave.;
  ..            . . .
  ..            . . .
  ..            . . .
```

(ONCE THE RECORD LOOKS GOOD, YOU HIT THE ATTENTION KEY TO LEAVE THE VISUAL EDITOR.)

```
: stop
-? update
-? stop
   . . .
   . . .
```

of changes to different staff members; the second if you only need to change one staff record.

5.4.1 Generating A List of Staff Members

In order to generate a list of staff members and their STAFF-ID numbers we will use the "DEFINE TABLE" command[2]. The following example gives the commands to generate this list.

```
$RUN *SPIRES
SELECT STAFF FILE
EXTRACT
SEQUENCE STAFF-NAME
DEFINE TABLE STAFF-ID(1,10) STAFF-NAME
USE -SAF
OUTPUT CLEAR
LIST OFF UNNUMBERED
STOP
```

An explanation of each step above follows:

1. **$RUN *SPIRES**

   This command tells MTS that we wish to enter Native SPIRES.

2. **SELECT STAFF FILE**

   We have three SPIRES sub-files which contain the information in DEACSS. We tell SPIRES that we wish to work on the staff information which is contained in the subfile "STAFF FILE".

3. **EXTRACT**

   The Extract command tells SPIRES that we want a set of pointers which contains a pointer to every record in the SPIRES subfile.

4. **SEQUENCE STAFF-NAME**

----

[2] A copy of the C.S. Write-up on DEFINE TABLE is included in Appendix G.

This tells SPIRES to sort our list of pointers in Alphabetical order based on the field "NAME".

5. **DEFINE TABLE STAFF-ID(1,10) STAFF-NAME**

Here we tell SPIRES that we want it to set up a Table in which to output specific fields from our records. In this case, we tell SPIRES that we want two fields, the STAFF-ID and the STAFF-NAME. We also specify that the STAFF-ID will be placed in columns 1-10, while the STAFF-NAME can take up the rest of the line.

6. **USE -SAF**

This command tells SPIRES that it is to use the file -SAF as the SPIRES Active file.

7. **OUTPUT CLEAR**

This command tells SPIRES to output the information requested for each of the records in the list of pointers. The information will be formatted as defined in the "DEFINE TABLE" command, and will be placed in the SPIRES Active File (in our case -SAF). The SPIRES Active File will be emptied before data is transferred.

8. **LIST OFF UNNUMBERED**

This command tells SPIRES to copy the contents of the SPIRES Active File to the printer. By default, this will go to the page printer and output will be returned to the Education Building. This command is identical to:

$COPY -SAF *PRINT*

9. **STOP**

This command gets us out of Native SPIRES and back to

MTS.

5.4.2 "FIND"ing an Individual STAFF-ID

In the STAFF FILE, the Staff members name is indexed. This means that we can use the SPIRES "FIND" command to get the records for all people with a certain name. If, for instance, we wished to find the record for a staff member by the name of Montgomerie, and to "TYPE" out that record, the commands would be:

```
# r *spires
# 20:54:16
 Spires active file set to: -SAF
 -Spires 6.0
 -Tracing on, 'EXPLAIN TRACING' for details
-? sel staff file
-? find n montgomerie
   -Result: 1 STAFF-FILE(S)
-? type

     STAFF-ID = 27;
     STAFF-NAME = Montgomerie, Thomas Craig, Mr.;
     STAFF-ACTIVE = YES;
   . . .
   . . .
   . . .
```

5.4.3 Updating the Staff Record

Once we have found out the STAFF-ID of the record to be modified, we can use Native SPIRES to transfer the record into the SPIRES Active File, use the MTS Editor to modify the record, then tell SPIRES to Update the record. Figure 8 shows the minimal information which is entered at the time a staff record is first generated using Interactive DEACSS. As well as this information, the following data fields should be entered for a Staff Member within the Department.

1. **STAFF-PREF-NAME**

    The name by which the staff member prefers to be known.

2. **STAFF-STREET**

    The street address of the staff member's home.

3. **STAFF-CITY**

    The city in which the staff member lives.

4. **STAFF-P-CODE**

    The Postal Code for the staff member's home address.

5. **STAFF-INTEREST**

    A list of interests for the staff member.

    Note: When editing the STAFF-INTEREST field, we use the "&" character where we wish to have the ";" character printed. This is because SPIRES expects the ";" character to indicate an end of a data element.

6. **STAFF-HOME-PHONE**

    This is the Home Phone number for the Staff member.

7. **FTE**

    The FTE field contains an Full Time Equivalent ranking for this staff member in the Department of Eduational Administration. This field is used in the STAFF WORKLOAD REPORT.

Figure 11 shows how we would use the SPIRES "FIND" command to find the STAFF-ID for Montgomerie, how we would modify his record using the Visual Editor and how we would "UPDATE" the record.

Figure 11

SAMPLE INTERACTION TO MODIFY A STAFF RECORD

```
# r *spires
# 20:54:16
 Spires active file set to: -SAF
 -Spires 6.0
 -Tracing on, 'EXPLAIN TRACING' for details
-? select staff file
-? find n montgomerie
   -Result: 1 STAFF-FILE(S)
-? type

    STAFF-ID = 27;
    STAFF-NAME = Montgomerie, Thomas Craig, Mr.;
    STAFF-ACTIVE = YES;
    STAFF-RANK = Sessional Lecturer (part time);
    DEPT = Department of Educational Administration;
    STAFF-OFFICE = 3-104;
    STAFF-PHONE = 432-3762;
-? trans 27 clear
-? edit
    Editing file -SAF
: v
```

(AT THIS POINT THE FOLLOWING WOULD APPEAR ON THE SCREEN)

```
1      STAFF-ID = 27;
2      STAFF-NAME = Montgomerie, Thomas Craig, Mr.;
3      STAFF-ACTIVE = YES;
4      STAFF-RANK = Sessional Lecturer (part time);
5      DEPT = Department of Educational Administration;
6      STAFF-OFFICE = 3-104;
7      STAFF-PHONE = 432-3762;
```

(YOU WOULD MODIFY THE RECORD USING THE VISUAL EDITOR SO THAT THE FOLLOWING APPEARED ON THE SCREEN)

```
1       STAFF-ID = 27;
2       STAFF-NAME = Montgomerie, Thomas Craig, Mr.;
3       STAFF-ACTIVE = YES;
3.1     STAFF-PREF-NAME = Craig;
3.2     STAFF-STREET = 11647 - 77 Ave.;
3.3     STAFF-CITY = Edmonton;
3.4     STAFF-P-CODE = T6G 0M4;
3.5     STAFF-INTEREST = computer applications to
3.6      education& research design and analysis.;
3.7     STAFF-HOME-PHONE = 432-2628;
4       STAFF-RANK = Sessional Lecturer (part-time);
5       DEPT = Department of Educational Administration;
6       STAFF-OFFICE = 3-104;
7       STAFF-PHONE = 3762;
8       FTE = .33;
```

(ONCE THE RECORD LOOKS GOOD, YOU HIT THE ATTENTION KEY TO LEAVE THE VISUAL EDITOR.)

```
: stop
-? update
-? stop
   . . .
   . . .
```

TECHNICAL REFERENCES

Culham, E. *The File Editor*. Edmonton: Department of Computing Services, The University of Alberta, 1979.

Jackson, G.R. *SPIRES/370 File Definition*. Edmonton: Department of Computing Services, The University of Alberta, 1977.

Jackson, G.R. *SPIRES Searching and Updating*. Edmonton: Department of Computing Services, The University of Alberta, 1978.

Jackson, G.R., Davis, A. *SPIRES User's Guide*. Edmonton: Department of Computing Services, The University of Alberta, 1976.

Senda, R.E. *SPIRES/370 Data Base Management*. Edmonton: Department of Computing Services, The University of Alberta, 1978.

TEXTFORM Group. *TEXTFORM Introduction*. Edmonton: Department of Computing Services, The University of Alberta, 1978.

TEXTFORM Group. *TEXTFORM Reference Manual*. Edmonton: Department of Computing Services, The University of Alberta, 1980.

TEXTFORM Group. *TEXTFORM User*. Edmonton: Department of Computing Services, The University of Alberta, 1980.

Webster, C.D. *MTS Commands*. Edmonton: Department of Computing Services, The University of Alberta, 1978.

APPENDIX B - A GLOSSARY OF TERMS

ACTIVE FILE: When SPIRES is requested to perform an ADD, UPDATE, SCAN, OUTPUT, etc. command, it takes (places) data from (into) a file. The file that is used is referred to as the ACTIVE FILE.

BATCH: A mode of computer operation in which the user sets up a series of computer operations in which all necessary information is specified. This series of operations are carried out by the computer without the supervision of the user.

DATA: A set of characters, words or signals to which a significance can be assigned. (Hussain, 1973, p. 81)

DATA BASE: A collection of data organized to facilitate maintenance, query, and/or reporting.

DATA BASE MANAGEMENT SYSTEM: A method of managing and manipulating the data in a DATA BASE.

DATA ELEMENT: a single data field in a record. eg. a student's name

DATA RECORD: all data related to a unique identifier. eg. a given student's record.

FILE: The FILE is the largest accessable unit in SPIRES. A FILE consists of a number of SUBFILES which contain related data, but which exist for specific purposes.

FORMAT: Once some records have been found, the data will be presented to the user in a particular form. The FORMAT capability allows the SPIRES programmer to set up nice looking data displays.

INDEX: Data fields can be used to create an index much like the index in a book. This enables SPIRES to quickly find the keyword desired and retrieve the records concerned.

INFORMATION: Selected data that have been processed to make them meaningful. (Hussain, 1973, p. 81)

INFORMATION MANAGEMENT SYSTEM: A frequently used synonym for a DATA BASE MANAGEMENT SYSTEM.

INTERACTIVE: A mode of computer operation in which the user interactes directly with the computer (usually by means of a computer terminal) as processing progresses.

GOAL RECORD: same as DATA RECORD.

KEY: the unique identifier of a record.

MANAGEMENT INFORMATION SYSTEM: A system which is designed to provide Information for Management Decision Making.

MTS: An acronym for Michigan Terminal System, the Operating System which controls the execution of jobs on the computer at the University of Alberta.

NATIVE SPIRES: The basic commands for manipulating a SPIRES data base which are provided by the SPIRES system.

OPERATING SYSTEM: The controlling program in a computer which controls the operation, scheduling and allocation of resources to all other programs in the system.

PROTOCOLS: SPIRES PROTOCOLS give SPIRES the full features of a standard programming language. PROTOCOLS allow interaction between the MTS operating system and the SPIRES data base.

REPORT: To aid the searcher in the printing of output, the REPORT option allows for page headings, footers, tabulations, etc.

SEARCH: telling SPIRES to look for something.

SEARCH RESULT: the DATA RECORDS found by SPIRES.

SEARCH TERM: a key word used to find records on the desired topic.

SEARCH LOGIC: when more than one search term is used, some means of logical connection between terms is needed. Logic is provided by the provision of the "<",">", "<=", ">=", "=", and "¬=" qualifiers.

SPIRES: The *S*tanford *P*ublic *I*nformation *RE*trieval *S*ystem. A generalized Data Base Management System designed to be accessed primarily from on-line applications. SPIRES is a self-contained system which allows the user full access to all the abilities of the system without the need of the usual alloting of resources by some centralized Data Base Administrator.

SUBFILE: The SUBFILE is the primary data base 'unit' It consists of a series of DATA RECORDS linked together by INDEX RECORDS. The SUBFILE contains all references for a particular application, but has the ability to reference data in other SUBFILES within the current FILE.

USER: A person authorized to enter data or commands to the computer system.

TIME SHARING: An operating system in which many tasks are in some state of execution in the system at the same time, with some master task allocating time and resources to each other task.

APPENDIX C - STRUCTURE OF DEACSS

This appendix shows the structure proposed for the current Department of Education Administration Computer Support System. The hierarchical structure of the file is demonstrated in the following manner:

1. There are three independent subfiles:

   a. The Grad Records File (the student file)

   b. The Course File

   c. The Staff File

   Data held in one subfile is totally independent of data held in another subfile unless linked by a pointer from one subfile to another subfile. For example, the data in the staff file is in no way related to the data in the student file, except that there is a pointer in the staff file to each student that that staff member advises or upon whose committee they serve.

2. Within each subfile there are data structures and data elements.

   a. Data elements are single values or strings which can not be further divided. For example, the sex, marital status or telephone number of a student. Data elements are indicated at the end of a line in the following description.

   b. Data structures take no actual value themselves, but indicate the hierarchical relationship of data elements and of other structures.
      In the Student File, for example, the Address structure keeps the elements of Street, City, Province, etc. together for a single address. If we had a student who lived in St. Albert and worked in Edmonton, we could quite conceivably have two addresses for that student. Without the ability of the structure to indicate which data elements are related to each other, we could have two street addresses, two cities, etc., and not know which street address referred to which city.

Two symbols have been used in the following section to make the file structure easier to read:

* Indicates that the element or structure may occur more than once.

• Indicates that the element should be placed in a index.

GRAD RECORDS - STRUCTURE

- Student Identification Number
- Name
  - Legal Name•
  - Former Surname*•
  - Preferred Name
- Date File Last Modified
- Department of Current Registration
- Address*
  - Street
  - City
  - Province, State
  - Country
  - Postal Code, Zip Code
  - Telephone
  - Date Address Verified
  - Address Type
- Social Insurance Number
- Birth Date
- Sex
- Marital Status
- Number of Dependents

:

GRAD RECORDS - STRUCTURE (Continued)

:

- Family* Member
  - Preferred Name
  - Relationship
  - Birth Date
  - Next of Kin
  - Address*
    - Street
    - City
    - Province, State
    - Country
    - Postal Code, Zip Code
    - Telephone
    - Date Address Verified
    - Address Type
- Citizenship
  - Country of Citizenship
  - Date
- Immigration
  - Immigration Status
  - Date Granted
  - Expiration Date
- Entrance* Examination
  - Examination Name
  - Mark
  - Date of Completion
- Teaching* Certificate
  - Issuing Authority
  - Type
  - Duration of Certificate
  - Date of Issue

:

GRAD RECORDS - STRUCTURE (Continued)

⋮

- Work* Experience
  - Name of Organization
  - Address
  - Position*
    - Job Title
    - Dates
    - Job Description
    - Education Only*
      - Specialization•
      - Student Type
- Educational* Experience
  - Degree•
  - Institution
  - Field of Specialization•
  - Year Completed
- Assistance*
  - Date Assistance Starts
  - Date Assistance Ends
  - Date Requested
  - Decision
  - Date of Decision
  - Funding Source
  - Supervising Department/Agency
  - Nature of Work
  - Supervisor
  - Bursary
  - Salary

⋮

GRAD RECORDS - STRUCTURE (Continued)

:

- Academic* Awards
  - Granting Agency
  - Award Type
  - Date of Application
  - Amount
  - Period of Award
  - Award Status
  - Award Obtained
- Publication*
  - Title of Publication
  - Co-author*
  - Publisher
  - Date of Publication
  - Pages
- Research* Project
  - Funding Agency
  - Co-researcher*
  - Amount of Funding
  - Dates
- Follow-up* Questionnaire
  - Type of Questionnaire
  - Date Sent
  - Date Received

:

GRAD RECORDS - STRUCTURE (Continued)

- Departmental* Reference
  - Date Requested
  - Date Sent
  - Organization Receiving Reference
  - Address
- Program* Registration
  - Degree•
  - Program Special.*
    - Program•
    - Date of Modification
  - Current Status*
    - Status•
    - Date of Modification
  - Enroll. Status*
    - Status•
    - Date of Modification
  - Expected Convocation Date
  - Comments*
    - Comment Still Applicable
    - Comment
  - Reference *
    - Name
    - Address
    - Date Requested
    - Date Received
    - Satisfactory

GRAD RECORDS - STRUCTURE (Continued)

- Trans. *
  - Institution
  - Address
  - Date Requested
  - Date Received
  - Satisfactory
- Date Admitted/Refused
- Discon- tinued*
  - Date Discontinued
  - Reason for Discontinuation
  - Date Program Restarted
- Residency
  - Residency Status●
  - Residency Period*
    - Date Begun
    - Date Ended
    - Status
- Candidacy
  - Date Scheduled
  - Result
- Oral
  - Date Scheduled
  - Result
- Advisor*
  - Staff ID
  - Advisor Name●
  - Date Appointed
  - Date Terminated
- Thesis Project
  - Title
  - Keyword*
  - Abstract

GRAD RECORDS - STRUCTURE (Continued)

- Committee Member*
  - Staff ID
  - Committee Member Name•
  - Dept./Org.
  - Member Type
  - Committee Type*
  - Date Appointed
  - Date Terminated
- U of A Course*
  - Course Number•
  - Course Comment
  - Year*
    - Year•
    - Session*
      - Session•
      - Section*
        - Section•
        - Mark*
          - Type
          - Date
          - Mark
          - Ruling
        - Credit Prg.
        - Status
        - Trans
          - Cat.
          - Fac.

GRAD RECORDS - STRUCTURE (Continued)

:

Transfer Course
- Institution
- Course Name
- Mark
- Credit Program
- Date Accepted
- Student Category
- Transfer Credits
- Comment
- Session Taken
- Course Equivalent*

COURSE FILE - STRUCTURE

- Course Number
- Course Description
- Course Type
- Year*
  - Year
  - Session*
    - Session
    - Section*
      - Section
      - Number of Credits
      - Time Offered
      - Type of Section
      - Section Description
      - Term Offered
      - Staff ID
      - Days Offered
      - Place Offered
      - Student File Pointer

STAFF FILE - STRUCTURE

- Staff ID
- Name
- Active/Inactive
- Special Remarks
- Preferred Name
- Street Address
- City
- Postal Code
- Interests
- Home Phone
- Rank
- Department
- Office Number
- Office Address
- Office Phone
- Full Time Equivalent
- Pointer to Student File

APPENDIX D - SUMMARY DATA ELEMENT DICTIONARY FOR DEACSS

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY

1. Student Identification Number *(IDNO)*
   The Student Identification Number assigned by the University is unique to each student. It will be used as the key to the student record.

2. Legal Name *(LEGAL-NAME)*
   The full name of the student as referenced in legal documents. Form to be:
   "LAST, FIRST MIDDLE ... ,PREFERRED TITLE"
   Where PREFERRED TITLE is one of: Mr., Mrs., Ms., Dr., etc.

3. Former Surname *(FORMER-SURNAME)* - may occur more than once.
   A surname by which this student may have been known (eg. Maiden name).

4. Preferred Name *(PREFERRED-NAME)*
   The name by which the student prefers to be known. (Nickname, short form of name, etc.).

5. Date of Last Modification of File *(MODDATE)*
   The date the file was modified last - automatically performed by the system.

6. Department of Last Registration *(DEPARTMENT)*
   The Department in which the student is currently registered, or from which was last graduated.

7. Address *(ADDRESS-STR)* - may occur more than once.

   a. Street *(S-STREET)*
      The apartment number, house number and street.

   b. City *(S-CITY)*

   c. Province, State, etc.*(S-PROVINCE)*

   d. Country *(S-COUNTRY)*

   e. Postal Code, Zip Code, etc. *(S-POSTAL-CODE)*

   f. Telephone Number *(S-PHONE)*

   g. Date address verified *(S-VERIFIED-DATE)*
      This will be the last date on which the address was verified. This will facilitate the keeping of the address current.

   h. Address Type *(S-ADDR-TYPE)*
      The type of address. This may include:

      1) Local address while at University

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

2) Mailing address while at university

3) Current Address after leaving university

4) Moved from this address - current address unknown.

8. Social Insurance Number *(SOCIAL-INSUR-NO)*
The Government of Canada Social Insurance Number (if assigned).

9. Birth Date *(BIRTHDATE)*

10. Birth Place *(BIRTHPLACE)*

11. Sex *(SEX)*

a. Male

b. Female

12. Marital Status *(MARITAL-STATUS)*

a. Single

b. Married

c. Separated

d. Divorced

e. Widow/Widower

13. Number of Dependents *(NUMBER-OF-DEP)*
The total number of dependents this student has (including spouse).

14. Family Member *(FAMILY-STR)* - may occur more than once.
The information on family members helps in decisions such as the granting of assistanceships on the basis of need.

a. Preferred Name *(F-PRE-NAME)*
The name of the family member - form to be:
"LAST, PREFERRED NAME, PREFERRED REFERENCE"

b. Relationship *(RELATIONSHIP)*

1) Wife

2) Husband

3) Son

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

4) Daughter

c. Birth Date *(F-BIRTHDAY)*
All dates will be stored in standard Canadian date form.
YY/MM/DD

d. Next of Kin *(NEXT-OF-KIN)*
Yes or No

e. Next of Kin Address *(NOFK-ADDR)* - may occur more than once. Only for Next of Kin.

1) Street *(R-STREET)*
The apartment number, house number and street.

2) City *(R-CITY)*

3) Province, State, etc. *(R-PROVINCE)*

4) Country *(R-COUNTRY)*

5) Postal Code, Zip Code, etc. *(R-POSTAL-CODE)*

6) Telephone Number *(R-PHONE)*

7) Date address verified *(R-VERIFIED-DATE)*
This will be the last date on which the address was verified. This will facilitate the keeping of the address current.

8) Address Type *(R-ADDR-TYPE)*
Two kinds of address for next of kin may be available.

a) A home address

b) A work address

15. Citizenship Status *(CITIZEN-STR)*
This contains the current citizenship of the student.

a. Country of Citizenship *(COUNTRY)*

b. Date Granted *(C-DATE)*

16. Immigration Status *(IMMIGRA-STR)*

a. Immigration Status *(IMMI-STATUS)*

1) Citizen of Canada

2) Landed Immigrant

3) Student Visa

b. Date Granted *(DATE-GRANTED)*

c. Date of Expiration *(EXPIRE-DATE)*
This will be applicable only in the case of student visa or refugee status.

17. Entrance Examination *(TEST-STR)* - may occur more than once.

a. Name of the Examination *(TEST-NAME)*

1) TOEFL

2) Miller Analogies

b. Mark *(TEST-MARK)*

c. Date of Completion *(TEST-DATE)*

18. Work Experience *(WORK-STR)* - may occur more than once.

a. Name of Organization *(ORG-NAME)*

b. Address *(WORK-ADDRESS)*

1) Street *(W-STREET)*
The office number, house number and street.

2) City *(W-CITY)*

3) Province, State, etc.*(W-PROVINCE)*

4) Country *(W-COUNTRY)*

5) Postal Code, Zip Code, etc. *(W-POSTAL-CODE)*

6) Telephone Number *(W-PHONE)*

7) Date address verified *(W-VERIFIED-DATE)*
This will be the last date on which the address was verified. This will facilitate the keeping of the address current.

8) Address Type *(W-ADDR-TYPE)*

a) A permanent work address

b) A temporary work address

c. Position *(POSITION-STR)* - may occur more than once.

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

1) Job Title *(JOB-TITLE)*

   a) Superintendent

   b) Principal

   c) Vice Principal

   d) Department Head

   e) Other Educational

   f) Other

2) Dates *(JOB-DATE)*
   The time period for which the job was held.

3) Job Description *(JOB-DES)*
   A short description of a job which may not be common terminology.

4) Special information Only of interest in Education Positions *(EDUCATION-ONLY)*

   a) Specialization *(EDUC-SPEC)* This may be any specific educational specializations such as:

      - Administration

      - Secondary Education

      - Elementary Education

      - Post-Secondary Education

      - Special Education

      - Guidance Counselling

   b) Type of Students encountered *(STUDENT-TYPE)*
      This is a further classification of the Specialization field. This may only be used for a few special cases of students, eg.:

      - Nursing Students

      - Gifted Students

19. Teaching Certificate *(TEACH-CERT-STR)* - may occur more than once.

   a. Issuing Authority *(ISSUE-AUTHORITY)*

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

The government or entity responsible for issuing the certificate.

b. Type *(TYPE-OF-CERT)*
The type of certificate - such as:

1) Secondary Certificate

2) Letter of Authority

c. Duration of Certificate *(T-DURATION)*
This will be either:

1) Permanent, or

2) the date at which the certificate must be renewed.

d. Date of Issue *(DATE-OF-ISSUE)*

20. Educational Experience *(EDUC-EXP-STR)* - may occur more than once.

a. Degree *(DEGREE-OBTAINED)*

1) Ph. D.

2) M. Ed.

3) B. Ed.

4) etc.

b. Institution *(INSTITUTION)*
The name of the institution granting the degree.

c. Field of Specialization *(SPECIALIZATION)*
This is used to differentiate the different degrees. Possible areas might be:

1) Nursing

2) Engineering

3) Elementary Education

4) Secondary Education

5) Special Education

6) etc. The institution granting the degree as coded in the CODE subfile.

d. Year Completed *(YEAR-COMPLETED)*
The year the degree was granted.

21. Assistance *(ASSISTANCE-STR)* - may occur more than once.
Student assistance provided by the Department.

a. Date Assistance Starts *(ASSISTANCE-START)*

b. Date Assistance Ends *(ASSISTANCE-END)*

c. Date Requested *(DATE-REQUESTED)*
The date the student requested assistance for the period?

d. Decision *(DECISION)*
The decision on assistance, may be:

1) Pending

2) Granted

3) Refused

e. Date of Decision *(DATE-OF-DECISION)*
The date the request was granted or refused.

f. Funding Source *(FUNDING-SOURCE)*

1) Faculty of Graduate Studies

2) Faculty of Education

3) Department of Educational Administration

g. Supervising Department/Agency *(SUPER-DEPT)*

1) Department of Educational Administration

2) Practicum

3) Department of Education

h. Nature of Work *(NATURE-OF-WORK)*
The nature of the work for the period. Possible types are:

1) Teaching.

2) Research assistance.

3) Practicum supervision.

i. Supervisor *(SUPERVISOR)*

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

The professor or other person who this student has been assigned to assist.

j. Bursary (*BURSARY*)
The amount of bursary payed the student for the period.

k. Salary (*SALARY*)
The amount of salary payed the student for the period.

22. Academic Awards (*ACADEMIC-AWARDS*) - may occur more than once.

a. Granting Agency (*GRANTING-AGENCY*)
The official name of the agency granting the award. Examples are:

1) The University of Alberta

2) The Social Sciences Research Council

3) Kellogg Foundation

b. Award Type (*AWARD-TYPE*)
The type of the award. Examples are:

1) Dissertation Scholarship

2) Sabbatical leave

c. Date of Application (*DATE-APPLY*)
The date the student applied for the award.

d. Amount (*AWARD-AMOUNT*)
The amount of the award.

e. Period of Award (*AWARD-PERIOD*)
The period which the award covers, coded YY/MM/DD - YY/MM/DD

f. Award Status (*AWARD-STATUS*)
The Status of the award:

1) completed

2) refused

3) Active

g. Award Obtained (*AWARD-OBTAINED*)
Whether the award was obtained or not.

23. Publication (*PUBLICATION-STR*) - may occur more than once.
A list of the student's publications

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

a. Title of Publication (*TITLE*)

b. Co-author (*CO-AUTHOR*) - may occur more than once.

c. Publisher (*PUBLISHER*)

1) If this is a book or monograph, the publisher.

2) If a journal article, the journal, volume, and number.

d. Date of Publication (*PUBLICATION-DATE*)

1) For a book - the year.

2) For a journal - the year and month

e. Pages (*PAGES*)

1) For a book - the number of pages.

2) For a journal - the page range.

24. Research Project (*RESEARCH-PRO-STR*) - may occur more than once.
Research projects (preferably those which were funded) which this person has undertaken.

a. Funding Agency (*FUNDING-AGENCY*)

b. Co-researcher (*CO-RESEARCHER*) - may occur more than once.

c. Amount of funding (*FUNDING-AMOUNT*)

d. Dates (*FUND-DURATION*)
The duration of the project.

25. Follow-up Questionnaire (*QUESTION-STR*) - may occur more than once.
Departments may send out questionnaires to follow the careers of students, or for other information. This structure enables the department to link the participation of the students in these questionnaires back to the master file.

a. Type of Questionnaire (*QUESTION-TYPE*) The title of the questionnaire.

b. Date Sent (*DATE-SENT*)

c. Date Received (*DATE-REC*)

26. Departmental Reference (*DEPT-REF-STR*) - may occur more than

once.
We maintain a record of all requests from students for a departmental reference.

a. Date Requested *(DATE-REQUESTED)*

b. Date Sent *(DATE-REF-SENT)*

c. Organization Receiving Reference *(ORD-SENT)*

d. Address *(ORG-ADDRESS)*

1) Street *(O-STREET)*
The office number, house number and street.

2) City *(O-CITY)*

3) Province, State, etc.*(O-PROVINCE)*

4) Country *(O-COUNTRY)*

5) Postal Code, Zip Code, etc. *(O-POSTAL-CODE)*

6) Telephone Number *(O-PHONE)*

7) Date address verified *(O-VERIFIED-DATE)*
This will be the last date on which the address was verified. This will facilitate the keeping of the address current.

8) Address Type *(O-ADDR-TYPE)*

a) A permanent work address

b) A temporary work address

27. Program Registration *(PROG-REG-STR)* - may occur more than once.
This is the complete record of the student for one degree. As long as a student is enrolled in one degree, this record will be in force. If a student completes a degree and moves on to another, a new Registration Period record will be initiated.

a. Degree *(DEGREE)*

1) Special Student

2) M. Ed.

3) Ph. D.

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

4) Ed. D.

b. Program Specialization *(PROG-SPEC)*
Any special program within the degree.

1) Program *(PROGRAM)* - may occur more than once.

a) Diploma

b) Administration Development Program

c) Teaching Skills Improvement Program

d) Non-Thesis

e) Thesis

2) Date of Modification *(DATE-PROG-MOD)*

c. Current Status *(CUR-STATUS-STR)*
The status of the student in the program. This may be:

1) Status *(CURRENT-STATUS)* - may occur more than once.

a) Application Pending

b) Application Refused

c) Admitted as Special Student

d) Granted Diploma

e) Qualifying Graduate Student

f) Candidate for Masters

g) Granted Masters

h) Provisional Candidate for Ph. D.

i) Candidate for Ph. D.

j) Granted Ph. D.

2) Date of Modification *(DATE-STATUS-MOD)*

d. Enrollment Status *(ENROLL-STR)*

1) Status *(ENROLL-STATUS)* - may occur more than once.

a) Full Time

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

b) Part Time

c) Discontinued

d) Completed

2) Date of Modification *(DATE-ENROL-MOD)*

e. Expected Convocation Date *(EXP-CONV-DATE)*

f. Comments *(COMMENTS-STR)* - may occur more than once.

1) Comment Still Applicable *(PRINT-COMMENTS)*
Is this Comment still in force (yes) or has it been cleared (no).

2) Comment *(COMMENTS)*
Some comment of particulat application to this particular registration. (Could be something like "Registration accepted conditional to successful completion of Ed. Adm. 511/512.)

g. Reference *(PROG-REF-STR)* - may occur more than once.

1) Name *(REF-NAME)*
The name of the reference - form to be:
"LAST, PREFERRED NAME, PREFERRED REFERENCE"

2) Address *(REF-ADDRESS)*

a) Street *(REF-STREET)*
The office number, house number and street.

b) City *(REF-CITY)*

c) Province, State, etc. *(REF-PROVINCE)*

d) Country *(REF-COUNTRY)*

e) Postal Code, Zip Code, etc. *(REF-POSTAL-CODE)*

f) Telephone Number *(REF-PHONE)*

g) Date address verified *(REF-VERIFIED-DATE)*
This will be the last date on which the address was verified. This will facilitate the keeping of the address current.

h) Address Type *(REF-ADDR-TYPE)*

3) Date Requested *(DATE-REF-REQ)*

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

4) Date Received *(DATE-REF-REC)*

5) Satisfactory *(REF-SATISFACTORY)*
Yes/No

h. Transcript *(TRANSCRIPT-STR)* - may occur more than once.

1) Institution *(TRANS-INST)*
The name of the institution

2) Address *(TRANS-ADDRESS)*

a) Street *(REF-STREET)*
The office number, house number and street.

b) City *(T-CITY)*

c) Province, State, etc. *(T-PROVINCE)*

d) Country *(T-COUNTRY)*

e) Postal Code, Zip Code, etc. *(T-POSTAL-CODE)*

f) Telephone Number *(T-PHONE)*

g) Date address verified *(T-VERIFIED-DATE)*
This will be the last date on which the address was verified. This will facilitate the keeping of the address current.

h) Address Type *(T-ADDR-TYPE)*

3) Date Requested *(DATE-TRANS-REQ)*

4) Date Received *(DATE-TRANS-REC)*

5) Satisfactory *(TRS-SATISFACTORY)*
Yes/No

i. Date Admitted / Refused *(DATE-ADM-REFUSE)*

j. Discontinued *(DISCONTINUED)* - may occur more than once. This structure occurs if the student discontinues the program (either permanently or temporarily.)

1) Date Discontinued *(DATE-DISCONT)*

2) Reason for Discontinuation *(REASON-DIST)*

3) Date Program Restarted *(DATE-RESTART)*

k. Residency *(RESIDENCY)*

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

1) Residency Status *(RES-STATUS)*>
The status of the residency can be:

a) Not yet begun

b) Begun but not yet completed

c) Completed

2) Residency Period *(RES-STATUS-STR)* - may occur more than once.

a) Date Started *(RES-DATE-BEGIN)*

b) Date Ended *(RES-DATE-END)*

c) Status *(RESIDENCY-STATUS)*
The status for each period can be:

- Not yet begun
- Begun but not yet completed
- Not Completed Satisfactorily
- Completed Satisfactorily

l. Candidacy *(CANDIDACY)*

1) Date Scheduled *(CANDIDACY-DATE)*

2) Result *(CANDIDACY-RESULT)*

a) Successful

b) Unsuccessful

c) Adjourned to YY/MM/DD

m. Oral *(ORAL-RESULT)*

1) Adjourned to YY/MM/DD

2) Successful

3) Unsuccessful

n. Date of Oral *(ORAL-DATE)*

o. Advisor *(ADVISOR-STR)* - may occur more than once.

1) Staff ID *(ADVISOR-ID)*

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

2) Name *(ADVISOR)*

3) Date Appointed *(DATE-APPOINTED)*

4) Date Terminated *(DATE-TERMINATED)*

p. Thesis/Individual Study Project *(THESIS-STR)*

1) Title *(THESIS-TITLE)*

2) Keyword *(THESIS-KEYWORD)* - may occur more than once.

3) Abstract *(THESIS-ABSTRACT)*

q. Committee Member *(COMMITTEE-STR)* - may occur more than once.

1) Staff ID *(COM-NAAME-ID)*

2) Name *(COM-NAME)*

3) Department/Organization *(DEPT-ORG)*

4) Member Type *(MEMBER-TYPE)*

a) Department Member

b) Non-department Member

c) External member

5) Committee Type *(COMMITTEE-TYPE)*

a) Candidacy

b) Thesis

6) Date Appointed *(COM-DATE-APPOINT)*

7) Date Terminated *(COM-DATE-END)*

28. U of A Course *(UA-COURSE-STR)* - may occur more than once.

a. Course Number *(UA-COURSE)*
Will be in standard university form: eg. EDADM511

b. Course Comment *(UA-COURSE-COM)*
Any comment on this course for this student.

1) Credit equivalent to EDADM 511/512.

2) EDADM506 taken to erase this failure.

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

3) Year Structure *(YEAR-STR)*

a) Year *(YEAR)*

b) Session Structure *(SESSION-STR)*

- Session *(SESSION)*
- Section Structure *(SECTION-STR)*
  - Session *(SECTION)*
  - Mark Structure *(MARK-STR)*
    - Mark Type *(UA-MARK-TYPE)*
      - EN - Enrolled
      - MA - Mark
      - RE - Remark
    - Mark Date *(UA-MARK-DATE)*
      The date the mark was recorded.
    - Mark *(UA-MARK)*
      The actual mark
    - Ruling *(UA-RULING)*
      A ruling if required. (Eg. S for Supplemental granted)
  - Credit Program *(CREDIT-PROG)* - may occur more than once.
    The program registration to which this course will be applied.
  - Course Status *(COURSE.STATUS)*
    - Enrolled
    - Completed
  - U of A Transfer Course *(UA.TRANS.CR)*
    Information on U of A courses which were transfered from some other faculty.
    - Transfer Category *(UA.TRAN.CAT)*
      The category of the student at the time this course was taken.
    - Faculty *(UA.TRAN.FAC)*

GRAD RECORDS - SUMMARY DATA ELEMENT DICTIONARY (Continued)

c. Transfer Course Structure *(TRANS-COURSE-STR)* Information on courses which were transferred from some other institution.

1) Institution *(TRAN-INSTITUTION)* The institution at which this course was taken.

2) Course Name *(TRAN-COURSE-NAME)* The official name of this course at the transferring institution.

3) Mark *(TRAN-MARK)* The mark achieved in this course.

4) Credit Program *(TRANS-CR-PROG)* The Program to which this course will be credited.

5) Date *(TRAN-DATE)* The date this course was approved for credit.

6) Student Classification *(TRANS-ST-CAT)* The Classification of the student at the transferring institution at the time this course was taken.

7) Course Equivalent *(COURSE-EQUI)* The University of Alberta course equivalent granted for this

COURSE FILE - SUMMARY DATA ELEMENT DICTIONARY

1. <u>Course Number</u> *(COURSE-17)*
   Will be in standard university form: eg. EDADM511

2. <u>Course Description</u> *(DESCRIPTION-17)*
   The official calendar description of the course.

3. <u>Course Type</u> *(COURSE-TYPE)*
   The kind of course this is, could be one of:

   a. Lecture

   b. Independent Study

   c. Independent Project

4. <u>Year Structure</u> *(YEAR-17-STR)* - may occur more than once.
   Since we are building a cumulative file, we order our course information chronologically. This procedure makes it easy to search data and later, if necessary, data can be archived more easily.

   a. <u>Year</u> *(YEAR-17)*
      The year in which the course was <u>begun</u>.

   b. <u>Session Structure</u> *(SESSION-17-STR)*

      1) <u>Session</u> *(SESSION-17)* The session in which the course was <u>begun</u>.
         May be one of:

         a) Winter

         b) Spring

         c) Summer

         d) Fall

      2) <u>Section Structure</u> *(SECTION-17-STR)*

         a) <u>Section</u> *(SECTION-17)* The section number of the course.

         b) <u>Number of Credits</u> *(NO-CR-17)*
            The number of credits given for this section.

         c) <u>Time Offered</u> *(TIME-17)*
            The time of day this section was offered. The 24 hour clock will be used, so a course offered from 6:00 P.M. to 8:50 P.M. would be designated 1800-2050.

d) <u>Type of Section</u> *(TYPE-17)*
May be one of:

i) Lecture

ii) Laboratory

iii) Seminar

e) <u>Section Description</u> *(SEC-DESC-17)*
Some courses (such as experimental courses or individual studies) may have different sections with different descriptions. This field is to take care of such situations.

f) <u>Term Offered</u> *(TERM-17)*
The term this section is offered. Can be on of:

i) 1 - for first term

ii) 2 - for second term

iii) F - for both terms

g) <u>Staff ID</u> *(STAFF-ID-17)*
The identification code number of the instructor teaching the section.

h) <u>Days Offered</u> *(DAY-17)*
Days of the week for which the course is offered are:

i) M for Mondays only

ii) T for Tuesdays only

iii) W for Wednesdays only

iv) R for Thursdays only

v) F for Fridays only

vi) S for Saturdays only

vii) MWF for Monday, Wednesday, Friday, etc.

i) <u>Place Offered</u> *(PLACE-17)*
Location of course:

i) Room number if on Campus

ii) City of town if off campus

j) Student File Pointer (*SMFPTR*) - may occur more than once.
A pointer to the individual record of each student enrolled in this section.

INSTRUCTOR FILE - SUMMARY DATA ELEMENT DICTIONARY

1. Staff Member Identification *(STAFF-ID)*
   The identification code number of the instructor. This is used for linking between the different files.

2. Staff Member Name *(STAFF-NAME)*
   The name of the instructor. This is always entered as:
   Montgomerie, Thomas Craig, Mr.

3. Active/Inactive *(STAFF-ACTIVE)*
   If the staff member is currently teaching in the department, this should be "YES", otherwise "NO".

4. Remarks *(STAFF-REMARKS)*
   Certain members of the department have special designations which are stored here, eg.:

   a. Department Chairman

   b. Director, Center for Post Secondary Education

   c. Visiting Professor, etc.

5. Preferred Name *(STAFF-PREF-NAME)*
   The name by which this staff member is known by others in the department.

6. Home Address *(STAFF-STREET)*

7. City *(STAFF-CITY)*
   The city in which the staff member lives.

8. Postal Code *(STAFF-P-CODE)*
   The Postal Code for the home address of the staff member.

9. Interests *(STAFF-INTEREST)*
   The academic interests of the staff member.

10. Home Phone *(STAFF-HOME-PHONE)*
    The home phone number of the staff member.

11. Staff Rank *(STAFF-RANK)*
    The rank of the instructor.

12. Department *(DEPT)*
    The department in which this staff member hold his appointment.

13. Office Number *(STAFF-OFFICE)*
    The office number of the staff member.

14. Work Address *(ADDRESS)*
    This is the work address for those who are not working at the U. of A.

15. Office Phone *(STAFF-PHONE)*
    The office phone number of the member.

16. Full Time Equivalent *(FTE)*
    The FTE equivalent in the Department.

17. Student File Pointer *(POINTER)* - may occur more than once.
    A pointer to the individual record of each student who this staff member advises, or upon whose committee he serves.

APPENDIX E - SAMPLE REPORTS PRODUCED BY DEACSS

DEA/S/01
Page 1

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
AARDVARK, ALAN A., MR.
STUDENT FILE: 808080
Date of Last Modification: Aug. 27, 1980

PERSONAL INFORMATION

Aardvark, Alan A., Mr. (Alan)

U of A Student ID: 808080

Social Insurance Number: 888 888 888

Born on Sept. 14, 1945 in Hoboken, New York

STUDENT ADDRESS INFORMATION

Permanent Address - Verified: July 10, 1980
999 Llama Road
Edmonton, Alberta, Canada
T6G 0V0

Phone: 499-9999

FAMILY INFORMATION

Current Marital Status: Married
Number of Dependents: 1

Name of Family Member: Aardvark, Agnes, Mrs.

Name of Family Member: Aardvark, Anthony, Mr.
Birthday: June 16, 1979

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
AARDVARK, ALAN A., MR.
STUDENT FILE: 808080
Date of Last Modification: Aug. 27, 1980

WORK HISTORY

Employed by E.P.S.B.

Business Address - Verified: July 10, 1980
10010-107A Ave.
Edmonton, Alberta, Canada
T4G 0N0
Phone: 429-5621

Job Title: Teacher

Job Description: Teaching

Position Held: 1974 to present
Educational Specialization: Teaching
Student Type: Grade 1-3

DEA/S/01
Page 3

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
AARDVARK, ALAN A., MR.
STUDENT FILE: 808080
Date of Last Modification: Aug. 27, 1980

EDUCATION HISTORY

Degree Obtained: B.Sc.
Institution: University of Alberta
Specialization: Education
Degree completed in 1967

TEACHING CERTIFICATES

Teaching Certificate Issued by ASTD
Type of Certificate: 999999
Duration of this certificate: Permanent
This certificate issued: Sept. 1, 1978

PUBLICATIONS

Title: The Engineer as a Teacher
Co-Author: Smith, W.P.
Date of Publication: July 1979
Publisher: The Education Digested
Number of Pages: 23-43

DEA/S/01
DEPARTMENT OF EDUCATIONAL ADMINISTRATION Page 4
AARDVARK, ALAN A., MR.
STUDENT FILE: 808080
Date of Last Modification: Aug. 27, 1980

MASTERS PROGRAM

Accepted in Thesis Program

Accepted as Candidate for Masters on July 10, 1980

Enrollment Status: Full Time on July 10, 1980

This comment still applicable.
This student would like to enter the Ph.D. program. He has been told that if he has an 8.0 average at Christmas, he will be recommended for acceptance to the Ph.D. Program with residency effective Sept. 1, 1980.

ADVISOR	APPOINTED	TERMINATED
Seger	July 10, 1980	

UNIVERSITY OF ALBERTA COURSES TAKEN TOWARDS MASTERS

COURSE	YEAR	SES	SEC	MARK	RUL.	DATE	STATUS / TRANSFER
EDPSY502	1976	Spr.	A3	9		07/10/80	Advanced Credit Special Student Education
EDADM561	1966	Fall	A2	2		02/02/80	Complete
EDADM599	1966	Fall	A2A			02/02/80	ENROLLED

TRANSFER COURSE TAKEN TOWARDS MASTERS

Course: Compsci 100 University: Athabasca
Student Grade was Credit when taken in 1978
Category (at time course taken) special student
April 15, 1980 - Assigned Course Equivalent to EDADM506
This course was taken befor the student realized that computers were a passing fad, and would never seriously affect education.

DEA/S/02

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
FULL TIME PH.D.
STUDENT ADDRESS LIST
Sept. 6, 1980

ANDRUKO, Myrna	P	7115 - 87 Ave. Edmonton, Alberta T6B 0L6	469-6085
ARMSTRONG, Lee	P	3504 - 112 St. Edmonton, Alberta T6G 2E0	637-4608
BARRINGTON, Gail	P	85 Hearthstone Edmonton, Alberta T6H 5E5	436-5728
BECKMAN, David	P	10909 - 33 A Ave. Edmonton, Alberta T6J 3C6	436-0041
BIRD, David	U	146 Michener Park Edmonton, Alberta T6H 4M4	437-1796
CAMPBELL, Catherine	P	P.O. Box 4302 Edmonton South, Alberta T6E 4T3	434-3880
COOPER, Jeanne	P	Box 42 Smoky Lake, Alberta T0A 3C0	656-3730 (B)/656-659
DANELIUK, Carl	B	9832 - 74 St. Edmonton, Alberta T6A 2X5 (Moved)	469-0209
HABINSKI, Aharon	P	#506, 11135 - 83 Ave. Edmonton, Alberta T6G 2C6	433-3690
HALL, Pete	U	431 Michener Park Edmonton, Alberta	436-9997
HANNAH, Kathryn	U	#200 Pemina Hall University of Alberta, Edmonton, Alberta	

DEA/S/03

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
FULL TIME PH.D.
STUDENT / ADVISOR LIST
Sept. 6, 1980

ANDRUKO, Myrna	Seger
ARMSTRONG, Lee	MacKay
BARRINGTON, Gail	MacKay
BECKMAN, David	Seger
BIRD, David	MacKay
CAMPBELL, Catherine	MacKay
COOPER, Jeanne	Bumbarger
DANELIUK, Carl	Bumbarger
HABINSKI, Aharon	Balderson
HALL, Pete	Fris
HANNAH, Kathryn	Friesen
JAFFRAY, Roy	Richards
JEFFERSON, Anne	Friesen
KAIDA, Lawrence	Miklos
KUNJBEHARI, Lalta	Ingram
LOEWEN, Robert	Bryce
MARCH, Milton	Miklos
MUTEMA, Alfred	MacKay
NGATIA, Peter	Fris
PETERS, Frank	Ratsoy
PORNSIMA, Direk	Ward
PRACHONGCHIT, Sanan	Bryce
REES, Ruth	McIntosh
SELLINGER, Gerald	Friesen

DEA/S/04

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
FULL TIME PH.D.
STUDENT ADDRESS & ADVISOR LIST
Sept. 6, 1980

ANDRUKO, Myrna	P	7115 - 87 Ave. Edmonton, Alberta T6B 0L6 Phone: 469-6085	Seger
ARMSTRONG, Lee	P	3504 - 112 St. Edmonton, Alberta T6G 2E0 Phone: 637-4608	MacKay
BARRINGTON, Gail	P	85 Hearthstone Edmonton, Alberta T6H 5E5 Phone: 436-5728	MacKay
BECKMAN, David	P	10909 - 33 A Ave. Edmonton, Alberta T6J 3C6 Phone: 436-0041	Seger
BIRD, David	U	146 Michener Park Edmonton, Alberta T6H 4M4 Phone: 437-1796	MacKay
CAMPBELL, Catherine	P	P.O. Box 4302 Edmonton South, Alberta T6E 4T3 Phone: 434-3880	MacKay
COOPER, Jeanne	P	Box 42 Smoky Lake, Alberta T0A 3C0 Phone: 656-3730 (B)/656-659	Bumbarger
DANELIUK, Carl	B	9832 - 74 St. Edmonton, Alberta T6A 2X5 (Moved) Phone: 469-0209	Bumbarger
HABINSKI, Aharon	P	#506, 11135 - 83 Ave. Edmonton, Alberta T6G 2C6 Phone: 433-3690	Balderson

DEA/S/05

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
ALL STUDENTS
STUDENTS WITH NO ADVISOR ASSIGNED
Sept. 6, 1980

IDNO	NAME	DEGREE
580517	Bosch, Melvin Joseph, Mr.	MASTERS
791934	Brick, Michael Robert, Mr.	DIPLOMA
		MASTERS

DEA/S/06

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
REQUESTS FOR ASSISTANCE WHICH BEGIN SEPTEMBER 1, 1980

Sept. 6, 1980

Aardvark, Alan A., Mr.
U of A Student ID: 808080
Social Insurance Number: 888 888 888
Degree: MASTERS
Program Specialization: Thesis
Current Status: Candidate for Masters
Enrollment Status: Full Time
Assistance requested for the period Sept. 1, 1980 - April 30, 1981

Employed by E.P.S.B.
Job Title: Teacher
Job Description: Teaching
Position Held: 1974 to present
Educational Specialization: Teaching
Student Type: Grade 1-3

DEA/F/01

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
STAFF LIST
Sept. 6, 1980

BALDERSON, J.H., DR.	Associate Professor	7-149	3689
BERGEN, J.J., DR.	Professor	7-109	3373
BRYCE, R.C., DR.	Professor	7-153	3681
BUMBARGER, C.S., DR.	Professor	7-133E	5868
BYRNE, T. C., DR.	Honorary Professor	7-133J	3691
CALDWELL, B., DR.	Research Ass't. Prof.	7-107	2734
ENNS, F., DR.	Professor	7-145	5868
FRIESEN, D., DR.	Professor	7-115	3690
FRIS, J., DR.	Associate Professor	7-133F	4905
GUE, L.R., DR.	Professor	7-151	4906
HODGSON, E.D., DR.	Professor	7-144	4906
HOLDAWAY, E.A., DR.	Professor	7-113 / Rm. 1-16 Univ. Hall	3690 5295
INGRAM, E.J., DR.	Professor	7-133K	3691
KONRAD, A.G., DR.	Professor/Coordinator of the Centre for the Study of Postsecondary Education	7-133G	3651
LOEWEN, ROBERT, MR.	Sessional Lecturer		
MACKAY, D.A., DR.	Professor	7-147	2073
MAGNAN, D., MR.	Research Asst. Prof.	7-110	3792
MCINTOSH, R.G., DR.	Professor	7-155	3681
MIKLOS, E., DR.	Professor	7-117	4916
MIREAU, LAURIE, DR.	Sessional Lecturer		

DEA/F/02

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
ADVISOR/COMMITTEE MEMBER LIST
Sept. 6, 1980

SEGER, J.E., DR. Department of Educational Administration

ADVISOR OF THE FOLLOWING STUDENTS

Armann, Donna	Candidate for Masters	Part Time	Thesis
Bratland, Allan	Candidate for Masters	Part Time	Thesis
Cooke, Terence	Candidate for Masters	Full Time	Non-Thesis
Foisy, Marie-Clare	Candidate for Masters	Full Time	Non Thesis
Greene, Myrna	Candidate for Masters	Full Time	ADP
Hong, Soon	Candidate for Masters	Part Time	Non-Thesis
Jiry, Ronald	Candidate for Masters	Part Time	ADP
Johnston, Ross	Candidate for Masters	Part Time	ADP
Kelly, James	Candidate for Masters	Part Time	Non-Thesis
Kim, Lim	Candidate for Masters	Full Time	ADP
Letain, John	Candidate for Masters	Full Time	ADP
McLaren, Sylvia	Candidate for Masters	Part Time	Thesis
Millar, Marilyn	Candidate for Masters	Part Time	ADP
Nasruddin, Sharon	Candidate for Masters	Part Time	Non-Thesis
Netzer, Margaret	Candidate for Masters	Part Time	Unknown
Podlubny, Ken	Candidate for Masters	Part Time	Non Thesis
Rozylo, Joan	Candidate for Masters	Part Time	ADP
Turnbull, Amelia	Candidate for Masters	Part Time	Thesis
Vinge, Maureen	Candidate for Masters	Withdrawing	Non Thesis
Young, Norma	Candidate for Masters	Full Time	Unknown
Alexander, Anne	Prov. Cand. for Ph.D.	Part Time	
Andruko, Myrna	Prov. Cand. for Ph.D.	Full Time	
Beckman, David	Prov. Cand. for Ph.D.	Full Time	
Brodie, Carl	Prov. Cand. for Ph.D.	Part Time	
Byrne, Paul	Prov. Cand. for Ph.D.	Part Time	
Danyluk, Joseph	Candidate for Ph.D.	Part Time	
Decoux, Bruce	Prov. Cand. for Ph.D.	Part Time	
Fennell, Brian	Prov. Cand. for Ph.D.	Part Time	
Gawreluck, Robert	Prov. Cand. for Ph.D.	Part Time	
Germsheid, Dick	Candidate for Ph.D.	Part Time	
Letourneau, Leo	Candidate for Ph.D.	Part Time	
Mailloux, Claire	Prov. Cand. for Ph.D.	Part Time	
Marshall, June	Prov. Cand. for Ph.D.	Part Time	
Montgomerie, Craig	Candidate for Ph.D.	Part Time	

COMMITTEE OF THE FOLLOWING STUDENTS

Harrison, Keith	Candidate for Ph.D.	Part Time
Symyrozium, Lloyd	Candidate for Ph.D.	Part Time
Taylor, Bill	Candidate for Ph.D.	Part Time

DEA/F/03

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
STAFF HOME ADDRESS LIST
Sept. 6, 1980

J.H. Balderson	10528 - 75 Ave.	T6E 1J4	436-1809
J.J. Bergen	11242 - 75 Ave.	T6G 0H3	434-1508
R.C. Bryce	Box 41, Site 5, R.R. 2 Sherwood Park	T8A 3K2	467-5144
C.S. Bumbarger	10910 - 51 Ave.	T6H 0L2	436-5110
T. C. Byrne	8303 - 138 St.	T5R 0E1	483-3025
B. Caldwell	3512 - 117B St.	T6J 1W2	435-0257
F. Enns	8320 - 117 St.	T6G 1R3	439-2886
D. Friesen	10807 - 41 Ave.	T6J 2P3	434-7910
J. Fris	481 Knottwood Rd. West	T6K 2V6	462-4875
L.R. Gue	(Thailand)		
E.D. Hodgson	4816 - 111A St.	T6H 3G6	434-3675
E.A. Holdaway	11463 - 48 Ave.	T6H 0C9	436-3042
E.J. Ingram	7404 - 157 St.	T5R 1Z9	487-1239
A.G. Konrad	12404 - 40 Ave.	T6J 0S6	435-1074
Robert Loewen	311 Michener Park	T6H 4M5	436-1585
D.A. MacKay	13107 - 63 Ave.	T6H 1R9	434-1293
D. Magnan	2403 - 89 Street	T6K 2Y8	462-2223
R.G. McIntosh	5712 - 144 St.	T6H 4H5	435-1221
E. Miklos	3110 - 115 St.	T6J 3H8	434-8045
Laurie Mireau	3922 - 76 St.	T6K 1V6	462-1512
T.C. Montgomerie	11647 - 77 Ave.	T6G 0M4	432-2628
M. Nixon	13816 - 88 Ave.	T6R 4J2	483-5735

DEA/F/04

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
ACADEMIC STAFF INTEREST LIST
Sept. 6, 1980

BALDERSON, J.H., DR. Main interests: technology and structure of educational organizations; research design and analysis.	7-149	3689
BERGEN, J.J., DR. (On Leave) Main interests: organization and governance of education; intergovernmental agencies in education.	7-109	3373
BRYCE, R.C., DR. Main interests: governance of education in Canada; supervision of educational personnel; organization theory.	7-153	3681
BUMBARGER, C.S., DR. Main interests: organizational development; educational planning; personnel practices.	7-133E	5868
BYRNE, T. C., DR. Honorary Professor	7-133J	3691
CALDWELL, B., DR. Main interests: educational finance; policy analysis; decision making; and development of administrative skills.	7-107	2734
ENNS, F., DR. Main interests: research and teaching in organizational behavior; personnel administration; the nature and validity of knowledge in educational administration.	7-145	5868
FRIESEN, D., DR. Main interests: research in organization theory and administrative behavior, school programs, and student personnel administration.	7-115	3690

DEA/F/05

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
INSTRUCTIONAL WORKLOAD
SUMMER SESSION 1979
Sept. 6, 1980

NAME OF INSTRUCTOR	FTE	RANK	COURSE	SEC	T	TYPE	ENR
Dr. J.J. Bergen	1.0	Prof.	EDADM551	50	1	Lecture	24
Dr. B. Caldwell	.33	R. Ass't. Prof.	EDADM505	50	1	Lecture	22
			EDADM545	50	1	Lecture	18
			EDADM592	70A	2	I. St.	1
Dr. J. Fris	1.0	Assoc. Prof.	EDADM461	55	2	Lecture	18
			EDADM501	60	2	Lecture	13
			EDADM502	60	2	Lecture	14
Dr. L.R. Gue	1.0	Prof.	EDADM591	70C		I. St.	1
			EDADM592	70B	2	I. St.	1
			EDADM901	70	2	I. Ass.	1
Dr. E.D. Hodgson	1.0	Prof.	EDADM553	55	2	Lecture	14
Dr. E.A. Holdaway	.33	Prof.	EDADM521	50	1	Lecture	18
			EDADM522	55	2	Lecture	6
			EDADM591	70D		I. St.	1
Dr. E.J. Ingram	1.0	Prof.	EDADM542	50	1	Lecture	20
Dr. A.G. Konrad	.33	Prof.	EDADM591	70A		I. St.	1
Dr. D.A. MacKay	1.0	Prof.	EDADM691	70	2	I. St.	1
Mr. Paul Mathews	1.0	Sess. Lec.	EDADM527	50	1	Lecture	19
Dr. E. Miklos	1.0	Prof.	EDADM501	50	1	Lecture	27
			EDADM502	55	2	Lecture	25
Dr. M. Nixon	1.0	Sess. Ass't. Prof.	EDADM463	55	2	Lecture	16
Dr. E.W. Ratsoy	1.0	Prof.	EDADM692	70	2	I. St.	1
Dr. D.M. Richards	1.0	Assoc. Prof.	EDADM591	70B		I. St.	1
Dr. J.E. Seger	1.0	Prof. (Chairman)	EDADM461	50	1	Lecture	21
			EDADM546	55	2	Lecture	17

APPENDIX F - EXAMPLE TEXTFORM USE WITH DEACSS

This appendix gives examples of how DEACSS format TEXT-LETTER can be interfaced with the text formatting language TEXTFORM. The Appendix includes:

1. an example of the output of TEXT-LETTER (the SPIRES Format which sets up a call to the TEXTFORM macro &NAME) for two students.

2. three examples of possible TEXTFORM macro &NAME which produce:

    a. a letter which tells students about the registration procedure for the coming year, and asks them to fill in and return a questionnaire,

    b. the questionnaire, which asks students to provide some information on their plans for the coming academic year and to verify departmental records, and

    c. an envelope.

3. The MTS jobs which produced the actual letters, questionnaires and envelopes.

4. Samples of the output from those three TEXTFORM runs.

SPIRES OUTPUT OF TWO STUDENTS USING FORMAT

TEXT-LETTER

```
<&NAME(' Mrs.',' H.L.','Montgomerie','11647 - 77 Ave.',-
'Edmonton, Alberta<NL>Canada','T6G 0M4','Heather',-
'Dr. E.J. Ingram','Phone: 436-2628','607 262 094',-
'640744','M.Ed.','Administrative Development Program')>
```

```
<&NAME(' Mr.',' T.C.','Montgomerie',-
'3-104 Education North, U of A',-
'Edmonton, Alberta<NL>Canada','T6G 2G5',-
'Craig','Dr. J.E. Seger',-
'Phone: 432-3762','606 268 894','631791','Ph.D.','  ')>
```

TEXTFORM SOURCE OF REGISTRATION LETTER

CONTAINED IN FILE SPIS:LETTER

```
<LAYOUT TUTORIAL('NOTOC','NOINDEX')>
<SIZE('REDUCE')>
<DEFINE MACRO &NAME>
<PNCTR = 0> <NPAGE> <NL 6, I L 11> June 13, 1980
<I L 0> <NL 3><PAR(1)> <PAR(2)> <PAR(3)>
<NL><PAR(4)> <NL><PAR(5)> <NL><PAR(6)>
<NL 2>Dear <PAR(1)> <PAR(3)>;
<NL 2> The Department of Educational Administration
requires information/confirmation relative to your
attendance plans for the 1980/81 Session.
Please complete the attached questionnaire and return it to
me by June 23, 1980.
<NL 2>I have indicated below our schedule for the Fall
Meetings prior to In-Person Registration.
Your advisor has been assigned to you as indicated on the
questionnaire.
Please see me if there are any discrepancies or errors.
<NL 2,F 3>AGENDA<NL 2 C,F 1>
Sept. 2<HS 3UN>10:00-11:00<I L 6>All-Full Time students
meet in Kiva
<I L 0>(Tuesday)<I L 6>(Second Floor, Education North).
<NL 2,I L 4>11:30<I L 6>All Full-Time Ph.D. Students
meet in Room¬7-152.
<NL>All Full-Time M.Ed. Students meet in Kiva.
<NL 2,I L 4>13:00<I L 6>Meet with advisor assigned and
set up program.
<NL 2,I L 0>Sept. 3-5<I L 4>In-Person registration
according to  published schedule.
<I L 0>
<NL 2> Sincerely,
<NL 3> J.E. Seger
<NL> Chairman
<NL 2>JES/jm <NL>Encl.
<ENDDEF MACRO &NAME>
$CONTINUE WITH NAMES
```

TEXTFORM SOURCE OF REGISTRATION QUESTIONNAIRE

CONTAINED IN FILE SPIS:QUEST

```
<LAYOUT TUTORIAL('NOTOC','NOINDEX')>
<SIZE('REDUCE')>
<DEFINE MACRO &USCORE20,REP(20,'_'),ENDDEF MACRO &USCORE20>
<DEFINE MACRO &UNKNOWN>
Please circle one of:
<NL,I L 5> 1) Thesis
<NL>2) Non Thesis
<NL>3) Administrative Development Program
<NL>4) Teaching Skills Improvement Program
<NL,I L 0>
<ENDDEF MACRO &UNKNOWN>
<DEFINE MACRO &NAME>
<PNCTR = 0,NPAGE>
<F 3>DEPARTMENT OF EDUCATIONAL ADMINISTRATION<NL C>
INFORMATION QUESTIONNAIRE<F 1,NL 2 C>
<F 2>PERSONAL INFORMATION:<F 1>
<NL 2,I B 2>If there are any errors or omissions in the
following information, please make the appropriate changes.
<I B 0>
<NL 2><PAR(1)> <PAR(2)> <PAR(3)> (<PAR(7)>)
<NL 2>Mailing Address:<I L 5,PAR(4)>
<NL><PAR(5)>
<I L 0>Postal Code:<I 1 5,PAR(6)>
<NL 2, I L 0,PAR(9)>
<NL 2>U of A Identification Number: <PAR(11)>
<NL 2>Social Insurance Number: <PAR(10)>
<NL 2>Program: <PAR(12)>
<PAR(13)>
<NL 2>Advisor Assigned: <PAR(8)>
<NL 3,F  2>ATTENDANCE PLANS:<F 1>
<NL 2>Will you be registering for the 1980-81 Session as a:
<NL 2,I L 1>Full-Time Student
(Part of Residency) <REP(8,'_')>
<NL>Part-Time Student
(Less than 3 courses per term) <REP(8,'_')>
<NL>Thesis Only: <REP(8,'_')> On Campus
<NL,HS 13UN,REP(8,'_')> Off Campus
<NL 2,I L 0>OR:<I L 1>Do you plan on having all course
requirements completed by Oct. 17, 1980 in order to
convocate this fall <REP(8,'_')>
<I L 0,NL 2>If you are registering for Thesis Only,
when do you anticipate completion:
<NL 2,REP(65,DASH),NL 2,REP(65,DASH),NL 2,REP(65,DASH)>
<NPAGE,F 2>ASSISTANTSHIP INFORMATION:<F 1>
<NL 2>Are you presently receiving an assistantship:
<NL,HS 10UN>Yes <REP(8,'_')><HS 5UN>No <REP(8,'_')>
<NL 2>Do you require an assistantship for the
1980-81 Session:
<NL,HS 10UN>Yes <REP(8,'_')><HS 5UN>No <REP(8,'_')>
```

```
<NL 2>If yes, please indicate which period(s) you would
require an assistantship:
<NL,I L 4>Fall: Sept.1, 1980 - Dec. 31, 1980 <REP(8,'_')>
<NL>Winter: Jan. 1, 1981 - April 30, 1981 <REP(8,'_')>
<NL>Intersession: May 1, 1981 - Aug. 31, 1981 <REP(8,'_')>
<I L 0,NL 2>
If you have any additional information or comments, please
indicate on the lines below:
<NL 2,REP(65,DASH),NL 2,REP(65,DASH)>
<NL 2,REP(65,DASH),NL 2,REP(65,DASH)>
<ENDDEF MACRO &NAME>
$CONTINUE WITH NAMES
```

TEXTFORM SOURCE OF ENVELOPE

CONTAINED IN FILE SPIS:ENVELOPE

```
<LAYOUT TUTORIAL('NOTOC','NOINDEX')>
<SIZE('REDUCE')>
<DEFINE MACRO &NAME>
<PNCTR = 0>
<NPAGE>
<NL 3><PAR(1)> <PAR(2)> <PAR(3)>
<NL><PAR(4)>
<NL><PAR(5)>
<NL><PAR(6)>
<ENDDEF MACRO &NAME>
$CONTINUE WITH NAMES
```

MTS JOB WHICH PRODUCES REGISTRATION LETTER

CONTAINED IN FILE SPIS:LETTERJOB

```
$SIGNON SPIS PRIO=L  RETURN=EDUC ROUTE=CNTR -
        PRINTER=PAGE FORMAT=FMTP2 FORM=1I OVERLAY=BLANK -
        PACKAGE=LOOSE P=300 T=15S 'BIN 6 CRAIG'
$RUN *TEXTFORM SCARDS=LETTER SPUNCH=-A
$RUN *PAGECONV SCARDS=-A PAR=P
$SIGNOFF
```

MTS JOB WHICH PRODUCES REGISTRATION QUESTIONNAIRE

CONTAINED IN FILE SPIS:QUESTJOB

```
$SIGNON SPIS PRIO=L  RETURN=EDUC ROUTE=CNTR -
        PRINTER=PAGE FORMAT=FMTP2 FORM=1J OVERLAY=BLANK -
        PACKAGE=LOOSE P=300 T=15S 'BIN 6 CRAIG'
$RUN *TEXTFORM SCARDS=QUEST SPUNCH=-A
$RUN *PAGECONV SCARDS=-A PAR=P
$SIGNOFF
```

MTS COMMANDS WHICH PRODUCE ENVELOPES

CONTAINED IN FILE SPIS:ENVELOPEJOB

```
$EMPTY -ENVOUT
$RUN NEW:TEXTFORM SCARDS=SUPENVELOPE  SPUNCH=-ENVOUT
$RUN *DITTO SCARDS=-ENVOUT
```

EXAMPLE LETTER TO H.L. MONTGOMERIE

(PRODUCED ON DEPARTMENTAL LETTERHEAD)

June 13, 1980

Mrs. H.L. Montgomerie
11647 - 77 Ave.
Edmonton, Alberta
Canada
T6G OM4

Dear Mrs. Montgomerie;

The Department of Educational Administration requires information/confirmation relative to your attendance plans for the 1980/81 Session. Please complete the attached questionnaire and return it to me by June 23, 1980.

I have indicated below our schedule for the Fall Meetings prior to In-Person Registration. Your advisor has been assiged to you as indicated on the questionnaire. Please see me if there are any discrepancies or errors.

AGENDA

Sept. 2 (Tuesday)	10:00-11:00	All-Full Time students meet in Kiva (Second Floor, Education North).
	11:30	All Full-Time Ph.D. Students meet in Room 7-152. All Full-Time M.Ed. Students meet in Kiva.
	13:00	Meet with advisor assigned and set up program.
Sept. 3-5	In-Person registration according to published schedule.	

Sincerely,

J.E. Seger
Chairman

JES/jm
ENCL.

EXAMPLE LETTER TO T.C. MONTGOMERIE

(PRODUCED ON DEPARTMENTAL LETTERHEAD)

June 13, 1980

Mr. T.C. Montgomerie
3-104 Education North, U of A
Edmonton, Alberta
Canada
T6G 2G5

Dear Mr. Montgomerie;

The Department of Educational Administration requires information/confirmation relative to your attendance plans for the 1980/81 Session. Please complete the attached questionnaire and return it to me by June 23, 1980.

I have indicated below our schedule for the Fall Meetings prior to In-Person Registration. Your advisor has been assiged to you as indicated on the questionnaire. Please see me if there are any discrepancies or errors.

AGENDA

Sept. 2 (Tuesday)	10:00-11:00	All-Full Time students meet in Kiva (Second Floor, Education North).
	11:30	All Full-Time Ph.D. Students meet in Room 7-152. All Full-Time M.Ed. Students meet in Kiva.
	13:00	Meet with advisor assigned and set up program.
Sept. 3-5	In-Person registration according to published schedule.	

Sincerely,

J.E. Seger
Chairman

JES/jm
ENCL.

EXAMPLE QUESTIONNAIRE TO H.L. MONTGOMERIE

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
INFORMATION QUESTIONNAIRE

PERSONAL INFORMATION:

If there are any errors or omissions in the following information, please make the appropriate changes.

Mrs. H.L. Montgomerie (Heather)

Mailing Address: 11647 - 77 Ave.
Edmonton, Alberta
Canada
Postal Code: T6G 0M4

Phone: 436-2628

U of A Identification Number: 640744

Social Insurance Number: 607 262 094

Program: M.Ed.- Administrative Development Program

Advisor Assigned: Dr. E.J. Ingram

ATTENDANCE PLANS:

Will you be registering for the 1980-81 Session as a:

Full-Time Student (Part of Residency) __
Part-Time Student (Less than 3 courses per term) __
Thesis Only: __ On Campus
__ Off Campus

OR: Do you plan on having all course requirements completed by Oct. 17, 1980 in order to convocate this fall __

If you are registering for Thesis Only, when do you anticipate completion:

___

___

___

EXAMPLE QUESTIONNAIRE TO H.L. MONTGOMERIE (Continued)

ASSISTANTSHIP INFORMATION:

Are you presently receiving an assistantship:
Yes __ No __

Do you require an assistantship for the 1980-81 Session:
Yes __ No __

If yes, please indicate which period(s) you would require an assistantship:
Fall: Sept.1, 1980 - Dec. 31, 1980 __
Winter: Jan. 1, 1981 - April 30, 1981 __
Intersession: May 1, 1981 - Aug. 31, 1981 __

If you have any additional information or comments, please indicate on the lines below:

____

____

____

____

EXAMPLE QUESTIONNAIRE TO T.C. MONTGOMERIE

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
INFORMATION QUESTIONNAIRE

PERSONAL INFORMATION:

If there are any errors or omissions in the following information, please make the appropriate changes.

Mr. T.C. Montgomerie (Craig)

Mailing Address: 3-104 Education North, U of A
Edmonton, Alberta
Canada

Postal Code: T6G 2G5

Phone: 432-3762

U of A Identification Number: 631791

Social Insurance Number: 606 268 894

Program: Ph.D.

Advisor Assigned: Dr. J.E. Seger

ATTENDANCE PLANS:

Will you be registering for the 1980-81 Session as a:

Full-Time Student (Part of Residency) __
Part-Time Student (Less than 3 courses per term) __
Thesis Only: __ On Campus
__ Off Campus

OR: Do you plan on having all course requirements completed by Oct. 17, 1980 in order to convocate this fall __

If you are registering for Thesis Only, when do you anticipate completion:

____

____

____

EXAMPLE QUESTIONNAIRE TO T.C. MONTGOMERIE (Continued)

ASSISTANTSHIP INFORMATION:

Are you presently receiving an assistantship:
Yes __ No __

Do you require an assistantship for the 1980-81 Session:
Yes __ No __

If yes, please indicate which period(s) you would require an assistantship:

Fall: Sept.1, 1980 - Dec. 31, 1980 __
Winter: Jan. 1, 1981 - April 30, 1981 __
Intersession: May 1, 1981 - Aug. 31, 1981 __

If you have any additional information or comments, please indicate on the lines below:

____

____

____

____

EXAMPLE ENVELOPES

Mrs. H.L. Montgomerie
11647 - 77 Ave.
Edmonton, Alberta
Canada
T6G 0M4

Mr. T.C. Montgomerie
3-104 Education North, U of A
Edmonton, Alberta
Canada
T6G 2G5

APPENDIX G - APPLICABLE DEPARTMENT OF COMPUTING SERVICES FREE PUBLICATIONS

This appendix contains some Department of Computing Services free publications which have been referenced. Included are:

1. R327.0280 Charging System

2. R275.0674 *DITTO

3. R331.1077 *TEXTFORM

4. R324.0780 9700 Page Printer with TEXTFORM

5. R223.0878 SPIRES DEFINE TABLE (DEF TAB) Command

Introduction

In 1970, General Faculties Council (GFC) approved, upon recommendation of the Computer Facilities and Policy Committee (CFPC) of GFC, a plan for charging for academic computing services. This plan has been in operation since fiscal year (April through March) 1971-72. This report outlines how the system works in general.

Charging for computing services is fairly typical of larger educational institutions with computing service centres. The purposes of charging are the following:

- establishment of a system of charging puts the costs of computing work into perspective with other institutional costs;

- the expenditures of the computing services organization (the Department of Computing Services at The University of Alberta) may be related to revenues received;

- in effect allocation of funds regulates the (principally central computer time) which a given individual or department may use--this is essential in a large institution where there are many users and where there will usually be a shortage of computing resources.

At The University of Alberta, there are two computing service organizations--the Department of Computing Services and the Office of Administrative Systems. Computing Services is not responsible for provision of computing services for administrative purposes. None of the descriptions below apply to the Office of Administrative Systems. The Office of Administrative Systems does not have a charging system in the usual sense, and is run on a completely different financial basis.

The Computer Facilities and Policy Committee (CFPC) of General Faculties Council

The CFPC has the following terms of reference:

1. To consider, and make recommendations on, applications for computing equipment throughout the University.

2. To develop and recommend to GFC policies for the effective operation and use of computing facilities.

3. To establish and maintain a charging system and allocate time for the use of computing facilities.

In this report, we are principally concerned with the third aspect of the Committee's responsibilities. Changes to the charging and budgeting system are first discussed and approved by the CFPC, and the Committee is directly involved each year in the allocation of computing funds.

However, under the first term of reference all proposals for major changes in computing equipment are discussed by the CFPC. The CFPC has delegated routine authority to Computing Services which is empowered to advise the Purchasing Department and the central administration when any acquisition of computing equipment is requested. Recommendations of Computing Services are reported to the CFPC.

With regard to the second term of reference of the CFPC, in practice the Committee usually restricts itself to major policy questions, leaving detailed operational questions to Computing Services. In principle Computing Services is responsible to the Committee in a policy sense and major policy questions are normally discussed thoroughly by the CFPC prior to implementation. Recommendations of the CFPC are communicated by its chairman, the Associate Vice-President (Academic), (Dr. W. F. Allen) to the appropriate agency: GFC for major policy changes, the central administration for financial recommendations. The Director of Computing Services sits on the CFPC and usually assumes responsibility for carrying out operational decisions of the Committee as they relate to academic matters.

Computing Services

Under the policy established by GFC, Computing Services charges for most work that it does. Each year Computing Services operating and capital budgets are negotiated directly with the central administration. When the budgets are determined, long range plans and policies which have been previously reviewed and approved by the CFPC are taken into account. Past and projected revenues received by the Department through charges for services are also considered. Thus the charging system becomes the principal means by which the level of Computing Services expenditures are regulated. Generally, the central administration is concerned with the Department's net operating budget, defined as the excess of the gross operating expenditures over revenues received. The net operating budget is a measure of the net direct cost of academic computer operations to the University.

Computing Services has authority to establish rates of charge. This is done under general policies formulated by the CFPC and approved by GFC. These general policies are as follows:

- rates should reflect the actual costs of providing service;

- rates should change smoothly over time, so that budgeting and planning by computer users are not unduly disrupted;

- rates should promote responsible and efficient use of computing facilities.

Computing Services also has the following objectives when setting rates:

- in general, rates should be related to the prevailing 'market cost' of computing services. This means that over time the cost of computing should decrease.

- Computing Services reviews revenues received as a measure of usage and relates them to departmental expenditures for various kinds of equipment and services, through an internal cost accounting system.

- Computing Services attempts to provide the maximum amount of service per dollar charged to the user consistent with long-run cost recovery.

How Computing Charges are Levied

In order to perform computing work the user opens a computing account with Computing Services. This involves filling in a Registration Form, available in the user's department, and sending it to Computing Services. The form is used to establish an identification code for computer use. When the computing account is opened, the user warrants that he will pay for any charges levied under that account, and identifies the source of funds. As work is performed, the computer automatically charges the account. At the end of each month, billing statements are sent to the user's department through the mail. In the case of University accounts, charges are automatically applied to the University budget account containing the source of funds for the work.

Control of the funds is entirely in the hands of the user. Once the user opens a computing account and authorizes expenditures of a certain maximum amount, he assumes responsibility for the account and any charges incurred through use of the account. It is often misunderstood that the University budget account, which is usually the source of funds for computing expenditures, and the computing account are totally different entities. It is up to the computer user to manage both. Comprehensive automatic facilities are available to the user to assist him in managing his account (the project ACCOUNTING facility). But it is up to the user to relate charges incurred under his computing account to funds which are available to him from the University or from grants, and which are contained in a University budget or trust account.

Sources of Funds for Computing Work

There are two major sources of funds for academic computing work. The first source is a research grant. When using grant funds the user is limited by the regulations of the granting agency and the general University regulations governing accounting of grant funds. Research grant funds are held in trust by the University in a trust account. Such an account will be billed automatically by Computing Services if the grant holder, who is usually the financial authority for a research grant trust account, so authorizes when he sets up his computing account.

Administration of computing charges to be paid from research grant trust accounts is usually simpler than University-supplied funds, because the research grant holder is in total control of the authorization for payment and the work performed.

The University also funds computing work. In fact most of the work performed at Computing Services is University funded. Such funds are placed in departmental accounts each fiscal year by the central administration on the advice of the CFPC. The departmental accounts are identified by the budget minor code 4520 (computing services) and are a part of the published University operating budget. Each such account is for the use of the department as a whole. Allocation of funds to an individual user from a departmental 4520 account must be authorized by the department chairman or by someone to whom the chairman has given financial authority. This has several implications:

- when Computing Services opens a computing account which will be paid from a departmental 4520 account (or any other University account) the authorization must be granted by the appropriate departmental authority.

- the financial authority for the 4520 account exercises control over all expenditures under that account. He usually suballocates funds to all computer users in the department who will be doing work to be paid from the departmental account.

- the CFPC and the central administration deal with the department as a whole when requests are made for computing funds. That is, additional 4520 funds can only be obtained by request of the departmental financial authority for all of the department's university computing funds.

Each year, the CFPC draws up a preliminary list of computing fund allocations for all departments. This is usually done in January. The Committee takes into account a) last year's allocation, b) whether last year's allocation was sufficient or not, c) budget transfers into and out of the departmental 4520 account, and d) any requests for additional funds which have been received.

These preliminary recommendations are circulated to the departments for scrutiny. After receiving any requests for alterations, the Committee draws up final recommendations which are forwarded to the Vice-President (Finance and Administration) for approval by the University Planning Committee, and ultimately by the Board. After approval, the amounts become part of the University operating budget, and are shown in the published budget as departmental 4520 budgets.

A certain amount of 'emergency computer funds' are budgeted and set aside each year by the CFPC. Departments may request access to such funds by writing the Secretary of the Committee. Requests are normally dealt with half way through the fiscal year, on 1 October, and on 1 January, although true emergency situations can be accommodated when the request is received. The Committee has delegated authority to deal with such emergency requests to its chairman, the Associate Vice-President (Academic) and its Secretary. These are presently Dr. W. F. Allen and Dr. D. H. Bent, Director of Computing Services, respectively.

When requesting additional funds, the department is expected to demonstrate that reasonable efforts are being made to conserve funds within the department, that additional funds are required from a department-wide point of view, and that alternative sources of funds are not available.

In the past, the CFPC has placed a higher priority on additional requests for the following:

- for departments previously having no computing funds or very small computing budgets;
- for teaching;
- for new faculty members or major new projects.

In general a department is expected to present requests for normal expenditures in January, when the preliminary budgets for the following fiscal year are drawn up.

Responsibility of the Department Chairman
(Dean, in case of non-departmentalized Faculties)

It can be seen from the above that the department chairman has the following responsibilities in connection with the funding of computing work:

- administration of the 4520 accounts;
- requests for any alterations in the 4520 accounts, and dealing with the CFPC or its officers in this regard.

The department chairman may delegate this responsibility (and normally does) to someone in the department who is more familiar with the computing work being done and the details of the computing accounts.

Most departments with large computing needs have set up departmental committees to coordinate the funding of computing work. In the case of smaller departments, there is usually an individual who has been made responsible for this area by the department head. Computing Services needs a clearly designated contact in each user department, and will give assistance where needed in explaining or administering the charging system.

Responsibility of the Computer User

The responsibilities of the user are:

- administer any computing accounts to which he has been permitted access

- arrange in his department for authorization to use departmental computing funds from a 4520 account

- if insufficient funds are available in the department to fund his computing work, inform the department chairman or designated financial authority to request additional funds in the following fiscal year, or in cases of emergency, emergency computing funds

- estimate his computing costs at least one fiscal year in the future

- make appropriate application for computing funds from research granting agencies if a research project involves computing work

- estimate and seek funds for any computing work to be carried out by a research assistant or student under his supervision.

Responsibilities of Computing Services

As indicated above, the main burden of financial responsibility falls on the user to arrange payment for computing work. Computing Services will, however,

- assist the user to set up his computing account;

- advise any departmental chairman or designated financial authority in administering computing accounts or departmental 4520 budgets;

- assist users to estimate the costs of computing projects when application is made for computing funds.

Some Technicalities Regarding University Computing Services (4520) Accounts

The use of departmental computing services (4520) budgets are, of course, governed by general regulations of the Comptroller. Some points of particular relevance are the following:
budget transfers out of 4520 accounts are normally not permitted. Authority to transfer 4520 funds to other accounts require the authorization of the Vice-President (Finance and Administration).
budget transfers into 4520 accounts are permitted, provided that the account the funds are transferred from allows such transfers. Computing Services will accept payment from any account for which financial authority has been given (Computing Services needs only an authorized signature on a Registration Form.)

Computing services (4520) funds are ineligible for fiscal year carry-over (flex budget), and therefore must be renewed annually.

For Further Information

For further information or assistance in administering any matter related to computing budgeting, Computing Services may be contacted. Usually the first point of contact is the Administrative and Information Services Manager, Mrs. Olga Kolar, 432-2261. For discussion of policy questions, or questions regarding additional University funding, the Director, Dr. Dale H. Bent may be contacted at 432-4767. The Director can also answer questions pertaining to the CFPC.

Contents: The object module of the ditto master preparation program.

Purpose: To convert output from *FMT or TEXT/360 to a format suitable for printing on a terminal such as an IBM 2741 or 1050. A typical application would be to produce ditto masters by inserting the masters in the terminal and having the output typed directly onto them.

Usage: The program is invoked by the $RUN command specifying *DITTO as the object file.

Logical I/O Units Referenced:

SCARDS - Line or sequential file containing the text input as produced by *FMT or TEXT/360.
SPRINT - Re-formatted output. Should remain defaulted to the terminal.
SERCOM - Prompting messages and error comments.

Description: This program produces output suitable for an IBM 2741, IBM 1050, or any of the newer terminals which operate at 300 baud and have an operational backspace character. If SPRINT is assigned to a device type other than the above the program will simply copy from SCARDS to SPRINT.

Before running *DITTO the user should issue the following device control commands (even if his terminal has a carriage width less than 255):

```
LEN=255
RM=255
```

This is because the output may consist of lines which contain backspace characters. The number of characters in the line may be greater than the carriage width but the text will take up the proper number of columns on the page.

When run the program will initially print the message:

```
TYPE "RETURN" WHEN READY FOR TYPED COPY
```

At this point, place the ditto master or blank piece of paper in the terminal, lined up one line above where the first line is to start. When ready, hit the RETURN key. The first page of the document will

be typed and the program will pause. Insert a new ditto master or blank piece of paper aligned as above and hit RETURN. The program will then type the second page. Continue as above for as many pages as desired.

The program will interpret the first character of all input lines as a carriage control character in the same manner as the line printer device support routines. Lines with carriage control characters which would cause them to be overprinted (e.g., for underlining) will be converted to a sequence of characters and back-space characters which will produce the same result on the terminal. Pages are identified by a "1" carriage control character (as produced by *FMT) or by a X'8B' carriage control (as produced by TEXT/360).

If trouble arises in the middle of a page, issue an attention interrupt. The program will rewind the SCARDS file to the beginning of the current page. After inserting a new ditto master, hitting RETURN will then repeat the page from the beginning.

The program terminates either after typing the last page in the SCARDS file or when the operator enters an end-of-file at the end of a page instead of hitting RETURN.

The program can also selectively scan for a particular page in the document. A page scan is requested by typing an integer number before hitting RETURN. This number is used to count the pages from the beginning of the SCARDS file. Note that this might <u>not</u> necessarily correspond to any page numbering which is printed on the document itself. That is, hitting "5" will type the fifth page from the beginning of the input file, regardless of the actual contents of that page. When using this feature it is advisable to first put a sheet of scrap paper into the terminal, type the desired page number and RETURN. The page will begin typing out and the operator can verify that it is the desired page. Then hit attention to go back to the beginning of the page, insert the ditto master and hit RETURN to type the final copy. After typing out a page found in this manner, the program will type the following page when RETURN is hit again.

Purpose: *TEXTFORM formats documents on the computer.

Contents: The object modules for the TEXTFORM system.

Usage: TEXTFORM is invoked by the MTS $RUN command.

Logical I/O Units Referenced:

SCARDS	-	contains text and commands to format the text.
SPRINT	-	is the proof output. The listing and errors are also written to this logical unit. See LIST and PROOF commands to suppress this output.
SPUNCH	-	the output document is written to the SPUNCH logical unit. See GALLEY command to suppress this output.
GUSER	-	the INTERACTIVE processes obtain the necessary commands from GUSER.
SERCOM	-	receives the INTERACTIVE messages generated. It also receives the error listing when TEXTFORM is run in terminal mode.

Other input/output units referenced: If you have a file named TXTF.PROLOG, TEXTFORM automatically processes commands (and text) from this file before processing from the SCARDS logical I/O unit.

Description: TEXTFORM accepts commands from the PAR= field of the $RUN command before processing commands and text from file TXTF.PROLOG or the logical I/O units. If you have a file named TXTF.PROLOG and do not want TEXTFORM to use this file, then the first command in the PAR= field must be NOPROLOG. If the LIST and CROSSREFERENCE commands are supplied in the PAR= field, the settings will remain in effect for the entire run and any other LIST or CROSSREFERENCE commands will be ignored.

Examples:

```
$RUN *TEXTFORM SCARDS=TEXTFILE SPUNCH=*PRINT*
```

In this example, the input is read from the file TEXTFILE and the output is sent to the printer.

```
$RUN *TEXTFORM SPUNCH=-OUTPUTFILE PAR=NOPROLOG
```

This example runs TEXTFORM without processing the text and commands in the file TXTF.PROLOG.

The Xerox 9700 is a high quality, fast printer. It prints on 8 1/2 x 11 inch sheet paper. The text may appear in landscape orientation (parallel to the 11 inch edge) or portrait orientation (parallel to the 8 1/2 inch edge).

An output device processor is available in NEW:TEXTFORM which allows the user to utilize the above features from TEXTFORM. To invoke the Xerox 9700 output device, use the TEXTFORM command

    <OUTPUTDEVICE 'X9700'>

The output directed through SPUNCH can be printed directly on the 9700 (*PRINT*) or written into a file for subsequent copying.

Currently, the TN character set with fonts normal (font 1), *italic* (font 2), and **bold** (font 3) are available on the X9700 output device. The TN character set can be used in portrait or landscape orientation. The APL character set with font 1 can only be used in portrait mode.

In order to change from portrait to landscape, redefine the page size; for example,

    <PAGESIZE=(11IN,8.5IN)>

The X9700 output device uses either LANDSCAPE.BLANK or PORTRAIT.BLANK overlays.

An X9700 proof device is also available. It enables the user to have SCARDS linenumbers associated with the X9700 output. With this proof device, text is printed in landscape format only. To invoke the X9700 proof device, use the TEXTFORM commands

    <OUTPUTDEVICE 'X9700',PROOFDEVICE 'X9700'>

When the X9700 proof device is in effect, no output is directed through SPUNCH.

Additional information about using the Xerox 9700 Page Printer is contained in the write-ups 'Xerox 9700 Page Printer' R332 and '*PAGECONV' R311.

The DEFINE TABLE command specifies items to be used to create a display of selected records in a subfile. The command may also specify where items are to appear in the display. In addition to the values of elements, "expressions" and "functions" can be displayed. In the following discussion, "item" refers to any expression, element, or function.

An expression is a simple equation specifying mathematical operations (add, subtract, multiply, divide) to be performed on the values of one or more elements in a single record; the value resulting from the evaluation of the expression is displayed for each record. For example, "PRICE*2" is an expression.

A function is a mathematical operation (sum, count, minimum, maximum, average, standard deviation) applied to the value of an element over several records; the value resulting from the evaluation of a function is displayed after all records have been displayed.

The basic form of the DEFINE TABLE command is:

    DEFINE TABLE item-list

where "item-list" is composed of element names (or their aliases), and expressions and functions. There can be no more than fifteen items. Any item in the list can optionally be followed by a specification of its position for output. If no position is specified, a default position is computed. Items should be specified in the order they are to appear in record display--if positions are not specified, then the items must be specified in order. Functions should be specified last.

The DEFINE TABLE command can be issued anytime after a subfile is selected. All records displayed after the DEFINE TABLE command is issued will use the tabular format specified. A subsequent DEFINE TABLE command will replace any previous one. A current DEFINE TABLE command can be cancelled or "undefined" by issuing the "ENDTABLE" command.

For example:

```
-? select restaurants
-? find cuisine chinese
-RESULT:  3 RESTAURANT(S)
-? define table name price phone
-? type

MAY 23, 1978                                        PAGE 1
NAME                 PRICE           PHONE
Feng Yuan            $3.00           494-7391
Jade Garden          $4.00           738-1228
China First          $4.25           326-3900

-? find cuisine italian
-RESULT:  5 RESTAURANT(S)
-? sequence name
-STACK:  5 RESTAURANT(S)
-? type

MAY 23, 1978                                   PAGE 1
NAME                    PRICE        PHONE
Barbarossa              $7.25        369-2626
Iron Pot                $6.00
Lorenzo
Pirro's Pizzeria        $2.75
Salerno                 $3.50

-? define table name address price
-? type

MAY 23, 1978                                         PAGE 1
NAME                ADDRESS                PRICE
Barbarossa          3003 El Camino Real $7.25
Iron Pot            Montgomery at Washi $6.00
Lorenzo
Pirro's Pizzeria    Shattuck Street        $2.75
Salerno             Shattuck               $3.50

-? endtable
-? type

ID = 53;
DATE-ADDED = JUNE 5, 1976;
DATE-UPDATED = JUNE 5, 1976;
NAME = Barbarossa;
CITY = Redwood City;
  STATE = California;
  PHONE = 369-2626;
  ADDRESS = 3003 El Camino Real;
```

Multiply Occurring and Non-Unique Elements

If a multiply occurring element or an element in a multiply occurring structure is named in a DEFINE TABLE command, then each occurrence of the element will be displayed starting on a new line, directly under previous occurrence.

For example, suppose LOCATION is a multiply occurring structure, with the singly occurring element CITY and the multiply occurring element PHONE in it. Then records with multiple occurrences of LOCATION may be displayed with the following commands:

```
-? find name china first
-RESULT:    1 RESTAURANT(S)
-? define table name city phone
-? type

MAY 23, 1978                                        PAGE 1
NAME               CITY             PHONE

China First        Palo Alto        326-3900
                                    326-3904
                   Menlo Park       326-3999
                                    326-4000
```

If an element name that is not unique in the goal record is specified in a DEFINE TABLE command, then the first occurrence of that element will be used. This situation will only arise when there are "floating" elements, which is quite rare.

Element and Expression Positioning

If no position information is specified for ANY items in the item-list on a DEFINE TABLE command, then default starting positions will be computed for you. Elements and expressions will be displayed in the order in which they were defined.

If all defaults are computed, then the system allows equal space for each element and expression in the list. The total width allowed for elements and expressions is determined by the LENGTH set for your terminal. Thus, if you SET LENGTH 72 (the default) and specify four elements and expressions on a DEFINE TABLE command, each gets a seventeen-character wide column: sixteen spaces for the item, and one blank to prevent it from interfering with the next item. So, in this case, elements or expressions longer than sixteen characters would be truncated at sixteen characters. For example, the following shows the defaults when four items are specified for a seventy-two column width:

```
-? set length 72
```

```
-? define table name address cuisine price
MAY 23, 1978                                                  PAGE 1
NAME            ADDRESS         CUISINE          PRICE
Seven Seas      443 Emerson     Chinese, America $2.75
Feng Yuan       3950 Middlefield Far-Eastern, Chi $3.00
Jade Garden     675 S. Bernardo Northern Chinese $4.00
China First     75 El Camino Rea Far-Eastern, Chi $4.25
Hunan           El Camino bet. W Far-Eastern, Chi $5.50
Panda           University Ave. Chinese,         $2.00
```

In general, when character items are displayed, they begin in the first character position in their column. When numeric information is displayed, it is usually "right adjusted" so that the last digit of all numbers for one item is in the same column. These defaults can be overridden.

You will probably not want to use all defaults if:

- you are specifying many elements and expressions for a small width
- you are specifying elements with lengthy values

If positions are specified for some items and not others, then the system will respect positions specified and default the others. This is not generally recommended. If in the previous example CUISINE is positioned explicitly in column 21, then NAME and ADDRESS, the two items preceding it, will be given columns 1-9 and 11-19.

Function Positioning

Since the results of function evaluation appear only in the last line of any tabular display, the results are positioned independently of elements and expressions. If no positions are specified for functions, the defaults are applied to them as for elements and expressions. That is, if two functions are specified and SET LENGTH 72 is in effect, each function gets thirty-five columns.

Headings

Headings are always supplied for the user; there is a default heading for each item that is displayed. In addition, the first line of the display always gives the current date (in "Month Day, Year" format) at the left margin and a page number at the right margin. The second line of each display gives headings for elements and expressions. If functions are specified, then the next to last line of each display gives the function heading; the last line gives the value of the function.

The default headings are always in uppercase, and are:

For ELEMENTS -- the name of the element or its alias, as specified in the DEFINE TABLE command.

For EXPRESSIONS -- the definition of the expression, with blanks removed, as specified in the DEFINE TABLE command.

For FUNCTIONS -- the definition of the function, with blanks removed, prefixed by the function name (SUM, COUNT, AVG, etc.).

If any heading is too long for the field displayed under it, the heading is truncated at the field length. Headings are always left adjusted; that is, they start in the first character position of the columns they head.

Carriage Control

While a DEFINE TABLE command is in effect, any records displayed into the active file (via the OUTPUT command or the IN ACTIVE prefix, for example) will have appropriate carriage control put on them. The carriage control can then be used when the active file is directed to the high-speed printer to give appropriate pagination.

Column one will always be either blank or have a "1" in it. The "1" tells the line printer to begin a new page at that point. This will make element and expression headings and the date and page number appear at the top of each page. The records and headings themselves will start in column two.

The information in the active file can be directed to the printer with a command such as:

```
PRINT NONUMBER CC
```

The "CC" indicates that carriage control is present. The system assumes a default of sixty lines per page. However, to change this, issue the command:

```
SET LINEPP number
```

Give the "number" of lines per page (LINEPP) you want. This command must be issued after the current DEFINE TABLE command was issued, and before a record display command that directs output to the active file.

Item Positioning

Sometimes the default positioning of items is not satisfactory. This is often the case when lengthy values are

being displayed, or when many values are being displayed into a narrow LENGTH. By specifying the positioning of items, you can created an aesthetically pleasing report.

You can specify:

- the starting position of an item
- the length per line of an item
- truncation or wrap-around for an item
- left (for text) or right (for numbers) adjustment of an item

Positioning information for each item is always enclosed in parentheses and placed after the item being positioned. The formal syntax is:

```
DEFINE TABLE item-name (start,length,adjust)
```

where:

"start" -- is the column in which the item is to begin.

"length" -- is the maximum length, on a single line, that the item can use before it is either truncated or wrapped to the following line. Truncation or wrapping is determined by "adjust."

"adjust" -- is either RIGHT (or R) or MARGINS (or M). RIGHT indicates that the value is to be moved to its rightmost margin, or "right adjusted." This allows one to align the last digits in numeric fields. MARGINS indicates that the value is to be wrapped around to a new line after every "length" characters. The new line will start in the same starting position as the first line. If MARGINS is not specified for "adjust," then values will be truncated at "length."

Some sample position specifications are:

```
NAME (10,25)
PRICE (40,6,RIGHT)
DESCRIPTION(50,20,MARGIN)
DATE (,8)
TIME(,8,R)
```

Note that blanks between an item's name and its position specification are optional. The effect of MARGINS and RIGHT adjustment is show in the following:

```
-? define table name(1,15) price(22,6,right)
comments(35,30,margins)
-? type

MAY 23, 1978                                                PAGE 1
NAME                   PRICE         COMMENTS

Sundance Mine          $10.00        The bar serves especially
good
                                     drinks, and is one of the
                                     place to meet interesting
                                     people.
Mekong Vietnam          $3.00        The Mekong has deceptive
                                     decor: it looks like a soda
                                     fountain, but the food is
                                     excellent.
Pirro's Pizzeri         $2.75        Friendly service and good
                                     value.  The food is just
fine.
The Echo                $6.94        Speciality is prime rib and
it
                                     is superb.
```

If a value is specified for "adjust," then a "length" must also be specified. The "start" position is optional unless MARGINS is used, but should usually be specified for best appearance. If no "start" is specified, then a comma should appear before the "length" specification, as the following examples show.

```
-? define table name(,20) price(,7,right) cuisine(,40)
-? type
MAY 23, 1978                                                PAGE 1
NAME                    PRICE   CUISINE

Sundance Mine            $7.94 American, Prime Rib
Mekong Vietnam Resta     $3.00 Far-Eastern, Vietnamese
Pirro's Pizzeria         $2.75 Continental, Italian, Pizza
The Echo                 $6.94 American
```

<u>Expression Specification</u>

In elaborate output formatting, you occasionally need to display a value that is mathematically derived from the values of one or more elements in a record. For instance, a salary database may have a monthly salary in it, and you need to display an annual salary; to do this, you need to multiply the salary value by twelve before displaying it. This kind of operation can be performed in the DEFINE TABLE command by

something called an "expression."

The DEFINE TABLE command will allow you to compute and display expressions. An expression is signalled by a period as its first character. Expressions can contain any combination of the four basic mathematical operations:

+ addition
- subtraction
* multiplication
/ division

However, expressions cannot contain parentheses, since a parenthesized position specification may follow an expression to indicate its placement in the output display.

For example, assume "MONTHLY.SALARY" is an element to be multiplied by twelve. An appropriate DEFINE TABLE command might be:

```
DEFINE TABLE NAME(1,30) .MONTHLY.SALARY * 12 (33,10,R)
```

The operators in an expression ("*" in this example) may optionally be surrounded by blanks if you wish.

Since "-" is the subtraction operator, element names that contain a "-" must be handled specially when used in expressions. The element name must be enclosed in quotation marks ("). (Note: apostrophes are not sufficient.) Thus, if the element name were MONTHLY-SALARY, an appropriate DEFINE TABLE command might be:

```
DEFINE TABLE NAME(1,30) ."MONTHLY-SALARY" * 12 (33,10,R)
```

More than one element can be named in an expression. As a more complicated example, suppose goal records contained an inventory item's unit cost (that is, cost for one item), and the total number of that item in stock. Assume the elements are called "UNIT.COST" and "QUANTITY." If you wanted to print the retail value of the current inventory, you might use a DEFINE TABLE command like the following:

```
DEFINE TABLE ITEM(1,10) .UNIT.COST * QUANTITY (15,10,RIGHT)
```

Note that expressions can only manipulate values in individual records. To perform calculations across records, you must use the function display capability of the DEFINE TABLE command.

Function Specification

When information in one record must be manipulated with information in other records (e.g., added to, averaged with, etc.), then functions must be used. For example, to find the

average price of the Chinese restaurants in the restaurant subfile, you would use the "AVG" (for "average") function. In a sense, functions allow you to perform mathematical operations on the columns of a table, while expressions allow you to manipulate the rows.

Each function has an object. In the above example, the object of the AVG function is the element PRICE. Functions can not only have elements as their object, but can also have expressions. The value of the object of a function must always be convertable to a numeric value.

The following describes the functions that are available:

SUM -- the object of the function is accumulated. A total across all records is the result.

COUNT -- the number of occurrences of the object is accumulated. A total number of occurrences of the object across all records processed is the result.

AVG -- the average value of the object across all occurrences of the object is the result.

MIN -- the smallest value of the object across all occurrences of the object is the result.

MAX -- the largest value of the object across all occurrences the object is the result.

STD -- the standard deviation (a measure of how close most values are to the average) of the object across all records processed is the result.

A function is specified in the DEFINE TABLE command by placing a period before one of the function names: e.g., .SUM, .COUNT, .AVG, .MIN, .MAX, .STD. The object of the function follows the function name, separated by a blank. Thus:

```
DEFINE TABLE .AVG PRICE
```

For example, to display the average price of some restaurants, you might do the following:

```
-? select restaurants
-? define table name phone price .avg price .count price
-? find cuisine chinese
-RESULT:   5 RESTAURANT(S)
-? type

MAY 21, 1978                              PAGE 1
NAME                        PHONE      PRICE

Seven Seas                322-4437     $2.75
Feng Yuan                 494-7391     $3.00
Jade Garden               738-1228     $4.00
China First               326-3900     $4.25
Panda                     322-4343     $2.00

AVG PRICE               COUNT PRICE
3.20                    5
```

Multiple functions can be specified, and functions can be followed by positioning information. For example:

```
DEFINE TABLE NAME(1) CUISINE(30) .AVG PRICE(20) .COUNT
PRICE(40)
```

Function objects can also be expressions, in which case the period that normally precedes an expression definition is not used. For example, to print the retail value of part stock and sum that value given only the elements UNIT.COST and QUANTITY, one might do the following:

```
DEFINE TABLE ITEM .UNIT.COST*QUANTITY .SUM
UNIT.COST*QUANTITY
```

APPENDIX H - STRUCTURE OF PROTOTYPE SYSTEM

This appendix shows the structure proposed for the Prototype Computer Support System. The hierarchical structure of the file is demonstrated in the following manner:

1. There are four independent subfiles:

   a. The Student File

   b. The Course File

   c. The Staff File

   d. The Mark File

   Data held in one subfile was intended to be totally independent of data held in another subfile. No linkage was set up between files (this was changed in the actual implementation.

2. Within each subfile there are data structures and data elements.

   a. Data elements are single values or strings which can not be further divided. For example, the sex, marital status or telephone number of a student. Data elements are indicated at the end of a line in the following description.

   b. Data structures take no actual value themselves, but indicate the hierarchical relationship of data elements and of other structures.
   In the Student File, for example, the Address structure keeps the elements of Street, City, Province, etc. together for a single address. If we had a student who lived in St. Albert and worked in Edmonton, we could quite conceivably have two addresses for that student. Without the ability of the structure to indicate which data elements are related to each other, we could have two street addresses, two cities, etc., and not know which street address referred to which city.

Two symbols have been used in the following section to make the file structure easier to read:

* Indicates that the element or structure may occur more than once.

• Indicates that the element should be placed in a index.

STUDENT FILE - STRUCTURE

- Student Identification Number
- Name
  - Legal Name●
  - Former Surname*●
  - Preferred Name
- Address*
  - Street
  - City
  - Province, State
  - Country
  - Postal Code, Zip Code
  - Telephone
  - Date Address Verified
  - Address Type
- Social Insurance Number
- Birth Date
- Sex
- Marital Status
- Family* Member
  - Preferred Name
  - Relationship
  - Birth Date
  - Next of Kin
  - Address (if Next of kin)*

:

STUDENT FILE - STRUCTURE (Continued)

:

- Citizenship/ Immigration
  - Citizen*
    - Country
    - Date
  - Immigra.*
    - Status
    - Date Granted
    - Expiration Date
- Miller Analogies
  - Required
  - Result*
    - Mark
    - Date of Completion
- TOEFL
  - Required
  - Result*
    - Mark
    - Date of Completion
- Work* Experience
  - Name of Organization
  - Address
  - Position*
    - Job Title
    - Dates
    - Job Description
    - Education Only*
      - Specialization•
      - Student Type
- Teaching* Certificate
  - Issuing Authority
  - Type
  - Duration of Certificate
  - Date of Issue

:

STUDENT FILE - STRUCTURE (Continued)

:

- Educational* Experience
  - Degree•
  - Institution
  - Field of Specialization•
  - Year Completed
- Assistance*
  - Period Covered
  - Date Requested
  - Decision
  - Date of Decision
  - Funding Source
  - Supervising Department/Agency
  - Nature of Work
  - Supervisor
  - Bursary
  - Salary
- Academic* Awards
  - Granting Agency
  - Award Type
  - Date of Application
  - Amount
  - Period of Award
- Publication*
  - Title of Publication
  - Co-author*
  - Publisher
  - Date of Publication
  - Pages

:

STUDENT FILE - STRUCTURE (Continued)

:

- Research* Project
  - Funding Agency
  - Co-researcher*
  - Amount of Funding
  - Dates
- Follow-up* Questionnaire
  - Type of Questionnaire
  - Date Sent
  - Date Received
- Departmental* Reference
  - Date Requested
  - Date Sent
  - Organization Receiving Reference
  - Address
- Program* Registration
  - Degree•
  - Program Special.*
    - Program•
    - Date of Modification
  - Current Status*
    - Status•
    - Date of Modification
  - Enroll. Status*
    - Status•
    - Date of Modification
  - Expected Convocation Date
  - Reference *
    - Name
    - Address
    - Date Requested
    - Date Received
    - Satisfactory

: :

STUDENT FILE - STRUCTURE (Continued)

- Trans. *
  - Institution
  - Address
  - Date Requested
  - Date Received
  - Satisfactory
- Date Admitted/Refused
- Reason for Refusal to Admit
- Discon-tinued*
  - Date Discontinued
  - Reason for Discontinuation
  - Date Program Restarted
- Residency
  - Residency Status•
  - Residency Period*
    - Dates
    - Status
- Candidacy
  - Date Scheduled
  - Result
- Oral
  - Date Scheduled
  - Result
- Advisor*
  - Name•
  - Date Appointed
  - Date Terminated
- Thesis/Project Title

STUDENT FILE - STRUCTURE (Continued)

:

- Committee Member*
  - Name•
  - Dept./Org.
  - Member Type
  - Committee Type*
  - Date Appointed
  - Date Terminated
- Course*
  - Course Number•
  - Credit* Program
    - Program Registration
    - Date of Modification
  - U. of A. Course
    - Year
    - Session
    - Section*
      - Type
      - Number
    - Pointer to Mark File
  - Transfer Course
    - Institution
    - Course Name
    - Mark
    - Transfer Credits
    - Course Equivalent*

COURSE FILE - STRUCTURE

- Course Number
- Weight
- Course Description
- Number of Hours per Week
- Section*
    - Section Number
    - Section Type
    - Instructor Code
    - Section Description
    - Year
    - Pointer to Mark File
    - Session

INSTRUCTOR FILE - STRUCTURE

```
┌──── Instructor Code
├──── Instructor Name
└──── Instructor Rank
```

MARK FILE - STRUCTURE

```
┌── Course Number:Year:Session:Section
│
│ Student*          ┌── Student ID•
└───────────────────┤
  Mark              ├── Mark*•
                    │
                    └── Ruling*•
```

APPENDIX I - SUMMARY DATA ELEMENT DICTIONARY FOR PROTOTYPE SYSTEM

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY

1. *Student Identification Number*
   The Student Identification Number assigned by the University is unique to each student. It will be used as the key to the student record.

2. *Name*
   The student name is broken into three parts:

   a. *Legal Name*
      The full name of the student as referenced in legal documents. Form to be:
      "LAST, FIRST MIDDLE ... ,PREFERRED TITLE"
      Where PREFERRED TITLE is one of: Mr., Mrs., Ms., Dr., etc.

   b. *Former Surname* - may occur more than once.
      A surname by which this student may have been known (eg. Maiden name).

   c. *Preferred Name*
      The name by which the student prefers to be known. (Nickname, short form of name, etc.).

3. *Address* - may occur more than once.

   a. *Street*
      The apartment number, house number and street.

   b. *City*

   c. *Province, State, etc.*

   d. *Country*

   e. *Postal Code, Zip Code, etc.*

   f. *Telephone Number*

   g. *Date address verified*
      This will be the last date on which the address was verified. This will facilitate the keeping of the address current.

   h. *Address Type*
      The type of address. This may include:

      1) Local address while at University

      2) Mailing address while at university

      3) Current Address after leaving university

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

4) Moved from this address - current address unknown.

4. *Social Insurance Number*
The Government of Canada Social Insurance Number (if assigned).

5. *Birth Date*
All dates will be stored in standard Canadian date form. YY/MM/DD

6. *Sex*

   a. Male

   b. Female

7. *Marital Status*

   a. Single

   b. Married

   c. Separated

   d. Divorced

   e. Widow/Widower

8. *Family Member* - may occur more than once.
The information on family members helps in decisions such as the granting of assistanceships on the basis of need.

   a. *Preferred Name*
   The name of the family member - form to be:
   "LAST, PREFERRED NAME, PREFERRED REFERENCE"

   b. *Relationship*

      1) Wife

      2) Husband

      3) Son

      4) Daughter

   c. *Birth Date*
   All dates will be stored in standard Canadian date form.
   YY/MM/DD

   d. *Next of Kin*
   Yes or No

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

e. *Address (if Next of Kin)* - may occur more than once. Two kinds of address for next of kin may be available.

1) A home address

2) A work address

9. *Citizenship/Immigration*
This will contain the complete citizenship and immigration record of the student.

a. *Citizen*

1) *Country of Citizenship*

2) *Date Granted*

b. *Immigration*

1) *Immigration Status*

a) Citizen of Canada

b) Landed Immigrant

c) Student Visa

2) *Date Granted*

3) *Date of Expiration*
This will be applicable only in the case of student visa or refugee status.

10. *Miller Analogies*

a. Is this *required* of the student?

b. *Result* - may occur more than once.

1) *Mark*

2) *Date of Completion*

11. *TOEFL*

a. Is this *required* of the student?

b. *Result* - may occur more than once.

1) *Mark*

2) *Date of Completion*

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

12. *Work Experience* - may occur more than once.

    a. *Name of Organization*

    b. *Address*
    Address will be structured identical to the personal address of the student.

    c. *Position* - may occur more than once.

        1) *Job Title*

            a) Superintendent

            b) Principal

            c) Vice Principal

            d) Department Head

            e) Other Educational

            f) Other

        2) *Dates*
        The time period for which the job was held.

        3) *Job Description*
        A short description of a job which may not be common terminology.

        4) Special information *Only* of interest in *Education* Positions

            a) *Specialization* This may be any specific educational specializations such as:

                - Administration

                - Secondary Education

                - Elementary Education

                - Post-Secondary Education

                - Special Education

                - Guidance Counselling

            b) *Type of Students encountered*
            This is a further classification of the Specialization field.

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

13. *Teaching Certificate* - may occur more than once.

    a. *Issuing Authority*
    The government or entity responsible for issuing the certificate.

    b. *Type*
    The type of certificate - such as:

    1) Secondary Certificate

    2) Letter of Authority

    c. *Duration of Certificate*
    This will be either:

    1) Permanent, or

    2) the date at which the certificate must be renewed.

    d. *Date of Issue*

14. *Educational Experience* - may occur more than once.

    a. *Degree*

    1) Ph. D.

    2) M. Ed.

    3) B. Ed.

    4) etc.

    b. *Institution*
    The name of the institution granting the degree.

    c. *Field of Specialization*
    This is used to differentiate the different degrees. Possible areas might be:

    1) Nursing

    2) Engineering

    3) Elementary Education

    4) Secondary Education

    5) Special Education

    6) etc. The institution granting the degree as coded in

the CODE subfile.

d. *Year Completed*
The year the degree was granted.

15. *Assistance* - may occur more than once.
Student assistance provided by the Department.

a. *Period Covered*

b. *Date Requested*
The date the student requested assistance for the period?

c. *Decision*
The decision on assistance, may be:

1) Pending

2) Granted

3) Refused

d. *Date of Decision*
The date the request was granted or refused.

e. *Funding Source*

1) Faculty of Graduate Studies

2) Faculty of Education

3) Department of Educational Administration

f. *Supervising Department/Agency*

1) Department of Educational Administration

2) Practicum

3) Department of Education

g. *Nature of Work*
The nature of the work for the period. Possible types are:

1) Teaching.

2) Research assistance.

3) Practicum supervision.

h. *Supervisor*

The professor or other person who this student has been assigned to assist.

i. *Bursary*
The amount of bursary payed the student for the period.

j. *Salary*
The amount of salary payed the student for the period.

16. *Academic Awards* - may occur more than once.

a. *Granting Agency*
The official name of the agency granting the award. Examples are:

1) The University of Alberta

2) The Social Sciences Research Council

3) Kellogg Foundation

b. *Award Type*
The type of the award. Examples are:

1) Dissertation Scholarship

2) Sabbatical leave

c. *Date of Application*
The date the student applied for the award.

d. *Amount*
The amount of the award.

e. *Period of Award*
The period which the award covers, coded YY/MM/DD - YY/MM/DD

17. *Publication* - may occur more than once.
A list of the student's publications

a. *Title of Publication*

b. *Co-author* - may occur more than once.

c. *Publisher*

1) If this is a book or monograph, the publisher.

2) If a journal article, the journal, volume, and number.

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

d. *Date of Publication*

1) For a book - the year.

2) For a journal - the year and month

e. *Pages*

1) For a book - the number of pages.

2) For a journal - the page range.

18. *Research Project* - may occur more than once.
Research projects (preferably those which were funded) which this person has undertaken.

a. *Funding Agency*

b. *Co-researcher* - may occur more than once.

c. *Amount of funding*

d. *Dates*
The duration of the project.

19. *Follow-up Questionnaire* - may occur more than once.
Departments may send out questionnaires to follow the careers of students, or for other information. This structure enables the department to link the participation of the students in these questionnaires back to the master file.

a. *Type of Questionnaire* The title of the questionnaire.

b. *Date Sent*

c. *Date Received*

20. *Departmental Reference* - may occur more than once.
We maintain a record of all requests from students for a departmental reference.

a. *Date Requested*

b. *Date Sent*

c. *Organization Receiving Reference*

d. *Address*

21. *Program Registration* - may occur more than once.
This is the complete record of the student for one degree. As long as a student is enrolled in one degree, this record will

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

be in force. If a student completes a degree and moves on to another, a new Registration Period record will be initiated.

a. *Degree*

1) Special Student

2) M. Ed.

3) Ph. D.

4) Ed. D.

b. *Program Specialization*

1) *Program* - may occur more than once.

a) Diploma

b) Administration Development Program

c) Non-Thesis

d) Thesis

2) *Date of Modification*

c. *Current Status*
The status of the student in the program. This may be:

1) *Status* - may occur more than once.

a) Application Pending

b) Application Refused

c) Admitted as Special Student

d) Granted Diploma

e) Qualifying Graduate Student

f) Candidate for Masters

g) Granted Masters

h) Provisional Candidate for Ph. D.

i) Candidate for Ph. D.

j) Granted Ph. D.

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

2) *Date of Modification*

d. *Enrollment Status*

1) *Status* - may occur more than once.

a) Full Time

b) Part Time

c) Discontinued

d) Completed

2) *Date of Modification*

e. *Expected Convocation Date*
Form to be: YY/MM/DD

f. *Reference* - may occur more than once.

1) *Name*
The name of the reference - form to be:
"LAST, PREFERRED NAME, PREFERRED REFERENCE"

2) *Address*
Address will be structured identical to the personal address of the student.

3) *Date Requested*
Form to be: YY/MM/DD

4) *Date Received*
Form to be: YY/MM/DD

5) *Satisfactory*
Yes/No

g. *Transcript* - may occur more than once.

1) *Institution*
The name of the institution

2) *Address*
Address will be structured identical to the personal address of the student.

3) *Date Requested*
Form to be: YY/MM/DD

4) *Date Received*
Form to be: YY/MM/DD

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

5) *Satisfactory*
Yes/No

h. *Date Admitted / Refused*
Form to be: YY/MM/DD

i. *Reason for Refusal to Admit*

j. *Discontinued* - may occur more than once.
This structure occurs if the student discontinues the program (either permanently or temporarily.)

1) *Date Discontinued*
Form to be: YY/MM/DD

2) *Reason for Discontinuation*

3) *Date Program Restarted*
Form to be: YY/MM/DD

k. *Residency*

1) *Residency Status*
The status of the residency can be:

a) Not yet begun

b) Begun but not yet completed

c) Completed

2) *Residency Period* - may occur more than once.

a) *Dates*
Form to be: YY/MM/DD - YY/MM/DD

b) *Status*
The status for each period can be:

- Not yet begun
- Begun but not yet completed
- Not Completed Satisfactorily
- Completed Satisfactorily

l. *Candidacy*

1) *Date Scheduled*
Form to be: YY/MM/DD

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

2) *Result*

a) Successful

b) Unsuccessful

c) Adjourned to YY/MM/DD

m. *Oral*

1) *Date Scheduled*
Form to be: YY/MM/DD

2) *Result*

a) Successful

b) Unsuccessful

c) Adjourned to YY/MM/DD

n. *Advisor* - may occur more than once.

1) *Name*

2) *Date Appointed*

3) *Date Terminated*

o. *Thesis/Individual Study Project Title*

p. *Committee Member* - may occur more than once.

1) *Name*

2) *Department/Organization*

3) *Member Type*

a) Department Member

b) Non-department Member

c) External member

4) *Committee Type*

a) Candidacy

b) Thesis

5) *Date Appointed*

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

6) *Date Terminated*

22. *Course* - may occur more than once.

a. *Course Number*
Will be in standard university form: eg. EDADM 511
Transfer courses will be assigned the code TRANS.

b. *Credit Program* - may occur more than once.
The program registration to which this course will be applied.

1) *Program Registration*

2) *Date of Modification*
The date on which the course was applied to this program registration.

c. *U. of A. Course*

1) *Year*
The year the course is offered.

2) *Session*
The session in which the course begins.

a) Fall

b) Winter

c) Spring

d) Summer

3) *Section* - may occur more than once.

a) *Type*

- Lecture

- Laboratory

- Seminar

- *Number*
The section number, usually consisting of one letter and one number.

4) *Pointer to Mark File*
This points to the record in the Mark File which contains the actual mark.

STUDENT FILE - SUMMARY DATA ELEMENT DICTIONARY (Continued)

d. *Transfer Course*

1) *Institution*
The institution giving the course.

2) *Course Name*
The course name assigned by the granting institution.

3) *Mark*

4) *Transfer Credits*
The number of credits allowed by the University of Alberta.

5) *Course Equivalent* - may occur more than once.
The course(s) for which this student has been granted transfer credit as a result of this course.
N.B. In some cases, transfer credit will be allowed, but not assigned as equivalent to a particular course or courses.

COURSE FILE - SUMMARY DATA ELEMENT DICTIONARY

1. *Course Number*
   Will be in standard university form: eg. EDADM 511

2. *Weight*
   The number of credits granted for the course.

3. *Course Description*
   The official calendar description of the course.

4. *Number of Hours per Week*
   The number of hours of instruction per week.

5. *Section* - may occur more than once.
   This information will be entered every time a section of the course is taught. Since it is possible that a course may have Lecture, Laboratory, or Seminar sections, these may all appear.

   a. *Section Number*
      The official calendar description of the section.

   b. *Section Type*
      May be one of:

      1) Lecture

      2) Laboratory

      3) Seminar

   c. *Instructor Code*
      The identification code number of the instructor teaching the section.

   d. *Section Description*
      Some courses (such as experimental courses or individual studies) may have different sections with different descriptions. This field is to take care of such situations.

   e. *Year*
      The year this section commenced.

f. *Pointer to Mark File*
A pointer to the record containing the marks of all students taking this course section.

g. *Session*
The Session in which this section is registered.

1) Winter

2) Summer

3) Spring

4) Fall

INSTRUCTOR FILE - SUMMARY DATA ELEMENT DICTIONARY

The instructor file will be a *temporary* structure designed to provide for the needs of the Student Record File. In a fully integrated Administrative Information System, the Instructor File would be much larger, with many searchable indices.

1. *Instructor Code*
   The identification code number of the instructor.

2. *Instructor Name*
   The name of the instructor.

3. *Instructor Rank*
   The rank of the instructor.

MARK FILE - SUMMARY DATA ELEMENT DICTIONARY

1. *Course Number:Year:Session:Section*
   Will be a composite of this information which will uniquely identify this section.

2. Student Mark Structure - may occur more than once.

   a. *Mark* - may occur more than once.
      Either a Stanine mark a valid two character indication of status

   b. *Ruling* - may occur more than once.

      1) Passed

      2) Failed

      3) Incomplete

APPENDIX J - STUDENT RECORD QUESTIONNAIRE AND COVERING LETTERS

This appendix contains copies of the information sent out at the beginning of the 1980-81 Winter Session to evaluate student attitudes towards DEACSS. Included are:

3. An example of the covering letter to all staff members asking for their cooperation in distributing the student records, letters to students and the questionnaires.

4. An example of the covering letter to students asking them to complete the questionnaire.

5. An example of the student questionnaire.

Sept. 2, 1980

Mr. T.C. Montgomerie
Sessional Lecturer (part-time)
Department of Educational Administration

Dear Craig,

As you know, our department has undertaken a project to computerize many departmental records. At this time, we have all our currently enrolled M. Ed. and Ph. D. student records in the system. We are attaching two copies of the complete student file for each of the students you advise. Please give one copy to the student and use the other for information you need while advising students for the 1980-81 session.

While every effort has been made to correct each student file, undoubtedly there are bound to be errors or ommissions (eg. our course marks only go back to 1977, and marks for Spring/Summer 1980 are not yet complete). The letter attached to each student file asks the student to correct their own file and forward any corrections to Betty. If you find the output of the computer files difficult to use, or if you feel information might be missing from any file, Betty will make the official file folders in the Departmental Office available to you.

If you have received files for students who you do not advise, or if you have not received files for students you do advise, please contact Betty at once so this can be rectified.

Thank you for your assistance.

Sincerely,

J.E. Seger
Chairman

JES/cz

Encl.

Sept. 2, 1980

Mr. T.C. Montgomerie
3-104 Education North, U of A
Edmonton, Alberta
Canada
T6G 2G5

Dear Mr. Montgomerie,

The Department of Educational Administration has undertaken a project to computerize many departmental records. Part of this project includes the computerization of your student record, including the information you have given the department and information the department maintains on each student (course marks, advisor, etc.).

You will find attached to this letter a complete copy of your computerized file. No information is stored in your student file which is not presented on the attached form. As this is the first time that this system has been used, there are bound to be errors or ommissions (eg. our course marks only go back to 1977, and marks for Spring/Summer 1980 are not yet complete). Would you please check the information presented and:

1. Correct any errors,

2. Add any missing information (except marks prior to 1977 and for Spring/Summer 1980).

3. Indicate any information you would like removed from the record, explaining why you would like it removed.

Once you have reviewed the attached file, would you please complete the short questionnaire which is attached. Your response will be kept in complete confidence.

Please return the questionnaire to Betty Lewis (Departmental Secretary) before Sept. 9, 1980. If you have any corrections, additions or deletions to your own student file, please give these to Betty at the same time.

Thank you for your assistance.

Sincerely,

J.E. Seger
Chairman

JES/cz

cc. Dr. J.E. Seger
Encl.

631791

DEPARTMENT OF EDUCATIONAL ADMINISTRATION
STUDENT RECORD QUESTIONNAIRE

A great deal of information is maintained in student record files. This information is used not only to indicate the status of individual students, but to provide information for planning and decision making.

The following questions will help the department evaluate attitudes towards student record files. While your input will not directly set policy, it will be used to guide us in the development of future policy in this area.

Section I

Please answer this section with reference to the information presented in your own student record. (For each question, please check all applicable individuals or groups.)

1. The following people should have access to <u>all</u> the information in my student file.

   a. Me (the student) _____ 10

   b. The Department Chairman _____ 11

   c. My Thesis/Program Advisor _____ 12

   d. An Instructor in a Course I am taking _____ 13

   e. Any Instuctor in the department _____ 14

   f. Any student in the department _____ 15

   g. The public at large _____ 16

2. The following people should have access to <u>the public</u> information (Name, Address, Telephone Number) in my student file.

   a. Me (the student) _____ 17

   b. The Department Chairman _____ 18

   c. My Thesis/Program Advisor _____ 19

   d. An Instructor in a Course I am taking _____ 20

   e. Any Instuctor in the department _____ 21

   f. Any student in the department _____ 22

   g. The public at large _____ 23

3. The following people should have access to

information on courses I have taken <u>and the marks obtained</u>.

a. Me (the student) ______ 24

b. The Department Chairman ______ 25

c. My Thesis/Program Advisor ______ 26

d. An Instructor in a Course I am taking ______ 27

e. Any Instuctor in the department ______ 28

f. Any student in the department ______ 29

g. The public at large ______ 30

4. The following people should have access to information on courses I have taken <u>but not the marks obtained</u> unless I have indicated so in question 3.

a. Me (the student) ______ 31

b. The Department Chairman ______ 32

c. My Thesis/Program Advisor ______ 33

d. An Instructor in a Course I am taking ______ 34

e. Any Instuctor in the department ______ 35

f. Any student in the department ______ 36

g. The public at large ______ 37

Section II

When answering these questions, please try to place yourself in the position of a chairman of a university department using the student file attached (your student file). Among other tasks, this job entails:

1) assigning student assistantships

2) planning future course offerings

3) writing letters of reference

4) recommending program extensions

5) course counselling

5. If you were a department chairman, what information would you like to see added to the student file. Please indicate to what use you would put this information. 38-49

____

____

____

____

____

6. If you were a department chairman, what information would you like to see deleted from the student file. Please indicate why you feel this information should be deleted. 50-61

____

____

____

____

____

Section III

Card 2

Now put yourself back in the role of a student.

7. As a student, what information would you like to see added to your student file. Please indicate why you think this information should be added. 10-21

____

____

____

____

____

8. As a student, what information would you like to see deleted from your student file. Please indicate why you feel this information should be deleted. 22-33

____

____

____

____

____

Section IV

9. Did your views as a department chairman conflict with those as a student?
   Yes _____ No _____ 34

If your views differed, we would like you to suggest an optimal position. (You may select one of your previous positions, or points from each position.)

10. What information should be added to the student file. Please indicate why you decided to add this information. 35-46

____

____

____

____

____

11. What information should be deleted from the student file. Please indicate why you decided this information should be deleted. 47-58

____

____

____

____

____

Section V

Finally we would like to assess your attitudes towards the storage of student records on a computer system.

12. As an administrator, do you feel that student records should be computerized?

Yes _____ No _____ 59

Why do you feel this way? 60-65

____

____

____

13. As an student, do you feel that student records should be computerized?

Yes _____ No _____ 66

Why do you feel this way? 67-72

____

____

____

14. Are your attitudes different towards the storage of student records in a computer system than they are towards the storage of the same records in the department's filing cabinets.

Yes _____ No _____ 73

If so, please explain the way(s)in which they differ. 74-78

____

____

____

15. Have you ever had a formal course concerning computers?

Yes _____ No _____ 79

16. Have you ever worked with a computer (other than simply receiving computer generated reports?)

Yes _____ No _____ 80